UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA E GESTÃO DO CONHECIMENTO

LUCIANO FRONTINO DE MEDEIROS

FRAMEWORK PARA ENGENHARIA E PROCESSAMENTO DE ONTOLOGIAS UTILIZANDO COMPUTAÇÃO QUÂNTICA

Tese de doutorado submetida ao Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento da Universidade Federal de Santa Catarina como requisito parcial para obtenção do grau de Doutor em Engenharia e Gestão do Conhecimento.

Orientadora: Profª. Drª. Lia Caetano Bastos
Co-Orientador: Prof. Dr. Rogério Cid Bastos

Florianópolis

2010

Catalogação na fonte pela Biblioteca Universitária
da
Universidade Federal de Santa Catarina

M488f Medeiros, Luciano Frontino de
Framework para engenharia e processamento de ontologias utilizando computação quântica [tese] / Luciano Frontino de Medeiros ; orientadora, Lia Caetano Bastos. - Florianópolis, SC, 2010.
204 p.: il., grafs.

Tese (doutorado) - Universidade Federal de Santa Catarina, Centro Tecnológico. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento.

Inclui referências

1. Engenharia e gestão do conhecimento. 2. Ontologia. 3. Algoritmos. 4. Computação quântica. I. Bastos, Lia Caetano. II. Universidade Federal de Santa Catarina. Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento. III. Título.

CDU 659.2

LUCIANO FRONTINO DE MEDEIROS

FRAMEWORK PARA ENGENHARIA E PROCESSAMENTO DE ONTOLOGIAS UTILIZANDO COMPUTAÇÃO QUÂNTICA

Esta tese foi julgada e aprovada para a obtenção do Grau de **Doutor em Engenharia e Gestão do Conhecimento no Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento** da Universidade Federal de Santa Catarina.

Florianópolis, 03 de setembro de 2010, 16:00 horas,
Auditório do EPS / UFSC.

_____
Prof. Dr. Roberto Carlos dos Santos Pacheco
Coordenador do Curso

Banca Examinadora:

_____ Prof. Dra. Lia Caetano Bastos Orientadora – UFSC	_____ Prof. Dr. Rogério Cid Bastos Co-orientador - UFSC
_____ Prof. Dr.Fernando Ostuni Gauthier Membro da Banca – UFSC	_____ Prof. Dr.Vinícius Medina Kern Membro da Banca - UFSC
_____ Prof.Dr.Ivan Gennadievitch Evseev Membro Externo – UTF-PR	_____ Prof. Dr. Paulo Afonso Bracarense Membro Externo- UFPR

_____
Prof. Dra. Mariana Grapeggia
Mediadora – Faculdade SENAC-SC

DEDICATÓRIA

Ao Supremo Arquiteto do Universo.

À minha família, Paula, minha esposa, e Amanda e Guilherme, meus filhos, pelo amor, paciência e tolerância das minhas ausências.

Aos meus pais, Osvaldo Frontino e Maria Delourdes de Medeiros, que foram os primeiros incentivadores e me ensinaram o valor do conhecimento.

A todos os mestres de minha caminhada, com os quais tive a oportunidade de conviver e aprender.

AGRADECIMENTOS

Gostaria de deixar meus agradecimentos a todos com quem tive oportunidade de conviver e que contribuíram de alguma forma com a elaboração deste trabalho.

Aos meus irmãos, Leonardo, Carolina e Leandro, os sobrinhos Larissa, Mateus e Pedro, os cunhados Gilberto e Nei, e cunhadas Márcia, Gesebel e Estela, meus sogros Nelson e Elsa, pela compreensão e pelo apoio.

À minha prima Vera Medeiros e meus primos Elton, Diogo, André, pelas diversas oportunidades e acolhidas em Florianópolis.

Ao amigo, irmão de ideias, professor e físico Hamilton Pereira da Silva, pelo incentivo e pela contribuição sobre os vários temas relacionados à Física Quântica.

Ao professor e físico Wilson Picler pelas oportunidades concedidas na criação da Faculdade Internacional de Curitiba, Fatec Internacional e Grupo Uninter.

Ao professor Arildo Dirceu Cordeiro, da UTF-PR e ao empresário Charles Stempniak, pela iniciação e incentivo na área de Ontologias.

Aos professores e colegas de convívio Elton Ivan Schneider, Marcelo Ribeiro de Paula Mascarenhas, Henrique José Castelo Branco, Glávio Paúra, Juliana Bergman, Roberto Seleme, Antônio Siemsen Munhoz, Benhur Gaio e às professoras Joana Romanowski, Iolanda Cortelazzo, Sandra Terezinha Urbanetz.

Ao amigo e colega Armando Kolbe Júnior e seus familiares pelo convívio, pronta disposição de suporte e as valiosas discussões.

Aos orientadores Lia Caetano Bastos e Rogério Cid Bastos, pelas valiosas contribuições e pela oportunidade ao processo de doutoramento.

Aos professores que participaram da Banca Examinadora desta tese e aos demais professores do Programa do EGC.

Aos amigos e colegas Sandro Rautenberg, Wagner Igarashi, Antonio Costa Gomes Filho, Charles Anderson Prada, Airton José dos Santos e Michele Andréia Borges.

Ao amigo e mentor João e a todos os integrantes de sua equipe.

Aos demais amigos e colegas do Grupo Uninter com quem tive oportunidade e conviver e compartilhar conhecimentos.

"*Conhecer é viver*"
(Humberto Maturana e
Francisco Varela)

RESUMO

MEDEIROS, Luciano Frontino de. **Framework para Engenharia e Processamento de Ontologias utilizando Computação Quântica**. 2010. 204f. Tese (Doutorado em Engenharia e Gestão do Conhecimento) – Programa de Pós-Graduação em Engenharia e Gestão do Conhecimento, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil, 2010.

Ontologias são recursos largamente utilizados para a representação de conhecimento em sistemas inteligentes. Ao longo do tempo, novos conhecimentos são adicionados e tais ontologias tendem a se tornar redes de complexidade crescente. Esta tese tem como objetivo trazer para a área da Engenharia Ontológica os benefícios de *performance* e representação que podem ser alcançados a partir do uso da Computação Quântica, a qual tem se mostrado vantajosa em áreas como a criptografia e buscas em conjuntos não ordenados. A abordagem é proposta a partir de um *framework* constituído dos seguintes conceitos derivados: superposição de classes, superposição de instâncias, superposição de relações e emaranhamento de classes. É demonstrado o uso de algoritmos quânticos para a superposição de classes e instâncias em ontologias, assim como aplicações sobre emaranhamento de classes. O trabalho também inclui um simulador para Computação Quântica como ferramenta de apoio na construção dos algoritmos, visualização dos circuitos quânticos e testes experimentais. A partir da ideia do armazenamento de estados superpostos por um tempo mais longo, o framework evolui para um modelo de representação de conhecimento em ontologias baseado no paradigma quântico. Sob esta ótica, são discutidas ramificações quanto à semelhança com o pensamento simbólico da mente humana e ainda o questionamento da própria definição de ontologias.

Palavras-chave: Ontologias, Engenharia do Conhecimento, Engenharia Ontológica, Processamento de Ontologias, Algoritmos para Ontologias, Computação Quântica.

ABSTRACT

MEDEIROS, Luciano Frontino de. **Framework for Ontological Engineering and Processing using Quantum Computing**. 2010. 204p. Thesis (Doctorate in Knowledge Engineering and Management) – Engineering and Knowledge Management Graduate Program, Universidade Federal de Santa Catarina, Florianópolis, SC, Brazil, 2010.

Ontologies are resources widely used for representing knowledge in intelligent systems. Through the years, new knowledge has been added and such ontologies tend to become more and more complex networks. This paper is focused on the benefits of performance and representation for the Ontologies Engineering area, which can be obtained from the use of the Quantum Computing concepts. This fact has been considerely advantageous in certain science computing areas, such as encryption and searching in unordered sets. The approach is proposed through a framework that shows the following derived concepts: superposition of classes, entanglement of classes, superposition of instances and superposition of relations. It is demonstrated the use of quantum algorithms for superposition of instances and classes in ontologies, as well as some possible applications in entanglement of classes. The study also includes a Quantum Computing simulator as a helping tool in building algorithms, visualizing quantum circuits and experimental testing. From the idea of storing the quantum states in a superposition for longer periods of time, the framework evolves to a representation model based on the quantum paradigm. Under this perspective, there are some considerations over branches towards the similarity with the human mind symbolic way of thinking and even considerations on the proper concept of ontologies.

Keywords: Ontologies, Knowledge Engineering, Ontological Engineering, Ontologies Processing, Algorithms for Ontologies, Quantum Computing.

ÍNDICE DE ILUSTRAÇÕES

Figura 1: Metodologia da tese de acordo com a pirâmide metodológica de Schreiber et al (2002). 28
Figura 2: Linguagens tradicionais de ontologias. 45
Figura 3: Linguagens de marcação de ontologias. 46
Figura 4: Integração de informação e extração de fontes de dados diversas usando ontologias. 48
Figura 5: Métodos básicos de Engenharia do Conhecimento e áreas de aplicação. 49
Figura 6: Linha do tempo com principais eventos relacionados ao desenvolvimento da Computação e Informação Quântica. 61
Figura 7: Esfera de Bloch com a representação de um q-bit. 62
Figura 8: Exemplos de portas digitais. 69
Figura 9: Circuito para a porta de troca. 73
Figura 10: Representação genérica de um circuito quântico. 74
Figura 11: Circuito quântico para produzir estados de Bell. 77
Figura 12: Circuito com 3 portas Walsh-Hadamard para gerar 8 estados superpostos. 80
Figura 13: Representação do algoritmo de Deutsch. A função $f(x)$ representa a transformação U_f. 82
Figura 14: Circuito simulador do algoritmo de Deutsch-Josza com três q-bits. 84
Figura 15: Circuito quântico para obter o estado de Bell β_{11}. 85
Figura 16: Ilustração do estado emaranhado referente ao singleto de Bohm. 86
Figura 17: Símbolo do operador de medida para um circuito quântico. 90
Figura 18: Exemplo de circuito para estimativa de fase. 101
Figura 19: Circuito para contagem quântica do exemplo. 103
Figura 20: Arquitetura híbrida de memórias para o processamento quântico de ontologias. 109
Figura 21: Arquitetura física para processamento. 109
Figura 22: Representação esquemática de uma memória quântica. 110
Figura 23: Ilustração da tarefa de validação de instâncias. 111
Figura 24: Diagrama esquemático do algoritmo de validação de instâncias. 113
Figura 25: Ilustração da tarefa de raciocínio transitivo. 115
Figura 26: Diagrama esquemático do algoritmo estocástico de raciocínio transitivo. 115
Figura 27: Ilustração da tarefa de *merging* de ontologias. 117

Figura 28: Algoritmo de verificação da existência de classes comuns para duas ontologias, utilizando contagem quântica. 118
Figura 29: Algoritmo para busca estocástica de classes comuns para duas ontologias, utilizando busca de Grover..... 118
Figura 30: Framework parcial, direcionada às tarefas de processamento de ontologias. 121
Figura 31: Ontologia de atendimento utilizada no algoritmo de validação. 122
Figura 32: Circuito para a leitura de memória e o oráculo do exemplo da ontologia de atendimento. 126
Figura 33: Circuito para contagem quântica do exemplo da ontologia de atendimento. 127
Figura 34: Ontologia para diagrama de influência utilizada como exemplo para o algoritmo de raciocínio transitivo. 128
Figura 35: Representação das 7 soluções transitivas dentre os 64 estados, após uma (a) e duas (b) iterações de Grover. 132
Figura 36: Circuito para a leitura de memória e o oráculo do exemplo da ontologia de raciocínio transitivo. 134
Figura 37: Exemplo para união de ontologias. 135
Figura 38: Representação da solução como classe comum dentre os 16 estados, após uma (a) e duas (b) iterações de Grover. 137
Figura 39: Circuito quântico para o algoritmo de *merging*. 139
Figura 40: Simulador de Computação Quântica desenvolvido para representação dos algoritmos quânticos. 140
Figura 41: Representação para três q-bits dos oito estados (2^3), equivalente à representação vetorial conforme a equação 13. 141
Figura 42: Mapa complexo de cores no padrão HSL coincidente com a representação de números complexos no plano *x*-*y* para o estado 141
Figura 43: Representação na esfera de Bloch do vetor $|0\rangle$. 142
Figura 44: Tela para visualização dos estados quânticos de acordo com a notação de Dirac. 142
Figura 45: A analogia entre o processo de formalização, o consenso e a medição quântica. 144
Figura 46: Representação de classes superpostas numa ontologia..... 145
Figura 47: Ontologia-exemplo de controle acadêmico em duas situações. 146
Figura 48: Ontologia-exemplo de controle acadêmico com classes superpostas. 147
Figura 49: Representação de superposição de relações, resultando em um conjunto específico de relações entre um determinado conjunto de classes. 148

Figura 50: Ontologia de exemplo para lógica de cenários refletindo duas percepções diferentes. 149
Figura 51: Superposição de relações para a ontologia-exemplo sobre lógica de variáveis. 150
Figura 52: Classes disjuntas representadas pelo estado emaranhado β_{01}. 151
Figura 53: Exemplo de lógica de variáveis para classes emaranhadas. 152
Figura 54: Circuito quântico para a produção da superposição do exemplo. 153
Figura 55: Evolução do *framework* mostrando agora as tarefas considerando a engenharia de ontologias. 154
Figura 56: *Framework* para o modelo CQ-Ontologias. 155
Figura 57: Circuito quântico para o *merging* de ontologias. 198
Figura 58: Exemplo de grafo para fechamento transitivo. 201

ÍNDICE DE QUADROS

Quadro 1: Diferentes conceitos de ontologia 32
Quadro 2: Categorias de ontologias. 38
Quadro 3: Princípios de construção de ontologias 41
Quadro 4: Metodologias para construção de ontologias. 44
Quadro 5: Exemplos de motores de inferência. 53
Quadro 6: Comparação entre Computação Clássica e Quântica. 67
Quadro 7: Portas quânticas dos Operadores de Pauli 70
Quadro 8: Operadores das portas *H*, *S* e *T* 70
Quadro 9: Operador CNOT 71
Quadro 10: Operadores adjuntos das portas *Y*, *S* e *T*. 72
Quadro 11: Operador de troca. 73
Quadro 12: Circuitos possíveis para os estados de Bell. 78
Quadro 13: Esquema para codificação superdensa. 88
Quadro 14: O algoritmo de Deustch escrito em QCL. 106
Quadro 15: O algoritmo de Deutsch na linguagem QML. 107
Quadro 16: Algoritmo genérico para validação de instâncias. 114
Quadro 17: Algoritmo genérico para raciocínio transitivo. 117
Quadro 18: Algoritmo genérico para merging de ontologias. 120
Quadro 19: Instâncias da ontologia de atendimento. 123
Quadro 20: Associação das instâncias da ontologia aos estados quânticos. 124
Quadro 21: Conversão das relações de influência em estados quânticos 129
Quadro 22: Conversão das classes em estados quânticos. 135
Quadro 23: Trabalhos recentes na área de hardware para computação quântica. 195
Quadro 24: Evolução dos estados do circuito de *merging* simulado. . 199
Quadro 25: Código em linguagem C do algoritmo de validação. 200
Quadro 26: Matriz de adjacência inicial. 201
Quadro 27: Matriz de adjacência de caminho dois. 202
Quadro 28: Código em linguagem C para o fechamento transitivo. ... 203
Quadro 29: Código em linguagem C do algoritmo de Warshall. 203
Quadro 30: Código em linguagem C do algoritmo de merging. 204

ÍNDICE DE SIGLAS E ABREVIAÇÕES

BPP Bounded Probability Polynomial
BQP Bounded Quantum Polynomial
CNOTControlled- NOT Gate (Porta NOT Controlada)
CHSHClauser-Horne-Shimony-Holt (Desigualdade)
CIE Commission Internationale d'Eclairage (padronização de sistemas de cores)
DAMLDARPA Markup Language
DFT Discrete Fourier Transform
DL Description Logic
EPR Einstein-Podolsky-Rosen (Experimento de Pensamento)
FFT Fast Fourier Transform
HTMLHyperText Markup Language
JTP Java Theorem Prover
NP Non-deterministic Polynomial
OIL Ontology Interchange Language
OWL Ontology Web-Language
PAL Protégé Axiom Language
POVMPositive Operator Value-Measured
PSM Problem Solving Methods
QED Quantum Eletrodynamics
QFT Quantum Fourier Transform
RDF Resource Description Framework
RDFS Resource Description Framework Schema
RMN Ressonância Magnética Nuclear
RSA Rivest-Shamir-Adleman – Algoritmo de Criptografia
SQUIDSuperconducting Quantum Interference Devices
UML Unified Modeling Language
XML eXtended Markup Language

SUMÁRIO

1 INTRODUÇÃO 16
1.1 CARACTERIZAÇÃO DO PROBLEMA 16
1.2 OBJETIVO GERAL 20
1.2.1 Objetivos Específicos 20
1.4 CONTEXTUALIZAÇÃO 20
1.5 MOTIVAÇÃO 25
1.6 ESCOPO DO TRABALHO 26
1.7 ASPECTOS METODOLÓGICOS 27
1.8 ADERÊNCIA AO PROGRAMA DE ENGENHARIA E GESTÃO DO CONHECIMENTO 28
2 FUNDAMENTAÇÃO TEÓRICA 30
2.1 ONTOLOGIAS 30
2.1.1 Conceitos de Ontologias 30
2.1.2 Componentes Principais de uma Ontologia 33
2.1.3 Tipos de Componentes de uma Ontologia 35
2.1.4 Categorização de Ontologias 37
2.2 CONSTRUÇÃO DE ONTOLOGIAS 38
2.2.1 Princípios de Construção de Ontologias 40
2.2.2 Metodologias para Construção de Ontologias 41
2.2.3 Linguagens de Ontologias 44
2.3 ONTOLOGIAS E ENGENHARIA DO CONHECIMENTO 46
2.3.1 Métodos Básicos e Técnicas 47
2.3.2 Mecanismos de Raciocínio e Inferência 50
2.3.3 O Consenso em Modelagem de Ontologias 53
2.4 COMPUTAÇÃO QUÂNTICA 54
2.4.1 Breve Histórico 54
2.4.2 O Q-Bit 59
2.4.3 Combinação de q-bits 63
2.4.4 Produto Interno e Ortonormalidade 67
2.4.5 Portas Quânticas 69
2.4.6 Circuitos Quânticos 73
2.4.7 Produto Tensorial e Registradores de Q-bits 75
2.4.8 Superposição e Interferência 80
2.4.9 Paralelismo Quântico 81
2.4.10 Emaranhamento 84
2.4.11 Medida 89
2.4.12 Descoerência 91
2.5 Algoritmos Quânticos 92
2.5.1 Algoritmo Quântico de Busca de Grover 94
2.5.2 Estimativa de Fase 100
2.5.3 Contagem Quântica 101
2.5.4 Simuladores de Circuitos Quânticos 104

2.5.5 Memória Quântica 104
2.5.6 Linguagens de Programação Quânticas 105
3 *FRAMEWORK* PARA ONTOLOGIAS E COMPUTAÇÃO QUÂNTICA CONSIDERANDO PROCESSAMENTO 108
3.1 VALIDAÇÃO DE INSTÂNCIAS 110
3.2 ALGORITMO PARA RACIOCÍNIO TRANSITIVO 114
3.3 *MERGING* DE ONTOLOGIAS 117
3.4 FRAMEWORK PARCIAL CONTEMPLANDO PROCESSAMENTO 120
3.5 CASOS PRÁTICOS DE APLICAÇÃO DOS ALGORITMOS 121
3.5.1 Caso Prático: Validação de Instâncias 122
3.5.2 Caso Prático: Raciocínio Transitivo 128
3.5.3 Caso Prático: *Merging* de Ontologias 133
3.6 SIMULADOR PARA COMPUTAÇÃO QUÂNTICA 139
4 EVOLUÇÃO DO *FRAMEWORK* PARA ASPECTOS DE ENGENHARIA 143
4.1 REPRESENTAÇÃO DE CLASSES SUPERPOSTAS 143
4.2 SUPERPOSIÇÃO DE RELAÇÕES 147
4.3 EMARANHAMENTO DE CLASSES 150
4.4 *FRAMEWORK* GERAL ONTOLOGIAS-CQ 154
4.5 CONSIDERAÇÕES 155
5 CONCLUSÃO E RECOMENDAÇÕES 159
5.1 CONCLUSÃO 159
5.2 RECOMENDAÇÕES PARA TRABALHOS FUTUROS 161
REFERÊNCIAS 163
APÊNDICE A - HARDWARE QUÂNTICO 185
APÊNDICE B – ALGORITMO SIMULADO DE *MERGING* 196
APÊNDICE C – ALGORITMOS CONVENCIONAIS 200
C.1 Algoritmo de Validação 200
C.2 Fechamento Transitivo 201
C.3 Algoritmo de Warshall 203
C.4 Algoritmo para Merging 204

1 INTRODUÇÃO

1.1 CARACTERIZAÇÃO DO PROBLEMA

A Engenharia e a Gestão do Conhecimento têm se beneficiado de muitas ferramentas oriundas de outras áreas, particularmente da Inteligência Artificial, para o alcance de seus objetivos. A intenção no uso estendido do poder da computação, imitando os processos de raciocínio e inferência humanos para obtenção de conhecimento relevante tem motivado a adoção de uma série de técnicas e ferramentas, como por exemplo, redes neurais artificiais, algoritmos genéticos, programação evolucionária e sistemas imunológicos artificiais.

De uma perspectiva mais simbólica do paradigma de Inteligência Artificial, as ontologias surgem da evolução das redes semânticas, baseadas na maneira em como a mente humana processa símbolos. Ontologias são definidas como uma "especificação formal de uma conceituação compartilhada" (GRUBER, 1993a). Esta conceituação geralmente encerra um conjunto de classes e relações entre classes, mostrando como um domínio de conhecimento está organizado. Esta organização de conhecimento tende a refletir o resultado de um consenso entre os indivíduos atores que compartilham tais conhecimentos (GÓMEZ-PÉREZ, 2004).

Entretanto, a construção de uma ontologia estará sujeita às diferentes percepções dos indivíduos que a constroem. Antes de um consenso, várias concepções sobre um domínio de conhecimento definido, ainda que contraditórias, estão presentes. O consenso direciona a formalização da ontologia, e as classes e relações explicitadas nesta ontologia irão representar da melhor forma os conceitos de tal domínio. As contradições relativas a uma visão principal ou eixo norteador da ontologia deixam de existir. Dependendo dos indivíduos e dos interesses, existe a possibilidade de uma ontologia resultar em algo que não irá representar fielmente o domínio estudado. Outro problema derivado disto são as mudanças inerentes ao domínio de conhecimento, requerendo que as ontologias sejam dinâmicas para continuar representando de forma relevante este domínio.

Somado a isto, ontologias tendem a aumentar em tamanho e complexidade, à medida que novos conhecimentos são acrescentados, em direção a uma representação melhor do domínio de conhecimento. Alguns exemplos ilustram esta situação, tais como a biblioteca de ontologias DAML (HORROCKS e VAN HARMELEN, 2001), que chegou a ter 282 ontologias, contendo em torno de 67.000 classes,

11.000 relações e 43.000 instâncias. A Protégé Ontology Library[1], mantida pela Stanford Center of Biomedical Informatics Research, possui em torno de 118 grupos de ontologias. A base de conhecimento Cyc, mantida atualmente pela OpenCyc.org, foi construída e também mapeada com outras ontologias ao longo de mais de 15 anos, a partir de 1 milhão de declarações entradas manualmente, contendo perto de 100.000 termos atômicos axiomatizados[2] (REED e LENAT, 2002). A WordNet, uma ontologia lingüística, atualmente na sua versão 3.0, contém cerca de 155.000 palavras, com 206.000 pares de palavras polissêmicas e 117.000 conjuntos de sinônimos[3].

Uma ontologia, ao longo do tempo, tende a tornar-se então uma rede de complexidade crescente, e a extração de informações relevantes começa a exigir algoritmos cada vez mais eficientes. Dependendo da atividade ou tarefa a ser executada, a complexidade algorítmica envolvida com redes ou grafos pode aumentar (CORMEN et al, 2002; TENENBAUM ET AL, 1995). Como exemplos de tais atividades, podem-se enumerar a **validação de instâncias**, **unificação** e **alinhamento** de ontologias, **raciocínio** ou **inferências** sobre ontologias e ainda o **aprendizado** de ontologias.

A organização de um domínio de conhecimento em uma ontologia requer que os dados sejam estruturados de maneira a não haver inconsistências, ou seja, na medida em que um axioma expresse uma verdade sobre tal domínio, esta verdade deve valer para todas as instâncias existentes nesta ontologia (GRUBER, 1993a). Instâncias que não atendam ao axioma deverão ser revistas ou então excluídas da ontologia. Tal atividade de **validação** de um axioma também pode se tornar dispendiosa com o crescimento de classes, relações ou instâncias da ontologia.

As ontologias criadas em domínios específicos e próximos entre si podem mais tarde requerer **unificação** (merging) ou **alinhamento**. Enquanto que no alinhamento interessa o mapeamento entre ontologias através de classes comuns, preservando-se estas ontologias originais, a unificação gera uma nova ontologia (NOY e MUSEN, 1999). Estas atividades começam pela busca das classes comuns às ontologias envolvidas, e dependendo da quantidade de classes e relações entre as mesmas, tal tarefa de busca pode ser dispendiosa, visto que a ontologia é estruturada tal como uma rede complexa.

---

[1] http://protege.cim3.net/
[2] http://sourceforge.net/projects/opencyc/
[3] http://wordnet.princeton.edu/wordnet/man/wnstats.7WN.html

Uma ontologia, após sua construção, também pode ser utilizada para tarefas de **inferências** ou **raciocínio**, com a possibilidade de se evidenciar novos conhecimentos (GÓMEZ-PÉREZ, 2004). Se uma classe "A" implica em outra classe "B", e tal classe "B" implica, por sua vez, em uma classe "C", uma inferência transitiva poderá evidenciar o conhecimento de que a classe "A" implicará na classe "C". Dependendo do tamanho, complexidade da ontologia e profundidade destas implicações, a busca de tais relacionamentos transitivos também pode se tornar bastante complexa.

O **aprendizado** de ontologias pode acontecer de maneira automatizada através de softwares ou agentes inteligentes. Ao longo do tempo, ontologias bastante extensas tendem a ser geradas. Os agentes identificam classes em bases de conhecimento não-estruturadas tais como textos, extração baseada em padrões, esquemas de bancos de dados ou ainda pela interoperabilidade entre sistemas (MAEDCHE e STAAB, 2000). Conforme o atendimento a critérios de relevância, o software insere as classes novas na ontologia, criando relações com classes já existentes. A busca destas classes iguais ou similares também deve ser um processo eficiente.

Portanto, buscou-se caracterizar duas situações-problema relevantes: uma relacionada à **engenharia** da ontologia, onde o resultado de um consenso entre várias concepções poderá influenciar na formalização de um modelo de ontologia; e a outra relacionada à **complexidade crescente** inerente a ela, precisando-se de algoritmos eficientes para o processamento e extração de informações a partir do que a ontologia representa.

A Computação Quântica e a Informação Quântica são áreas relativamente novas de conhecimento, que têm explorado o potencial existente no mundo quântico para o processamento de informações. Estas informações são representadas de forma diferente, indo além da representação binária clássica dos bits, adotando o conceito de **q-bit** ou **qubit** (NIELSEN e CHUANG, 2005). O q-bit possui uma enorme capacidade para a representação de informação. Porém, o aspecto mais relevante e de maior contribuição além da própria capacidade seriam os conceitos de **superposição** (que permite a existência simultânea de informações em um q-bit até que se faça a medida ou observação), e o **emaranhamento** (que possibilita aos q-bits manter associação não local, do tipo EPR). O emaranhamento entra em choque direto na Física com o conceito de localidade da Relatividade Especial, por prever correlações entre objetos quânticos através de ação não local e

instantânea, mas é um efeito já comprovado por inúmeros experimentos (ALBERT e GALCHEN, 2009).

A partir dos conceitos de superposição e emaranhamento, o poder de representação da informação em um meio quântico supera substancialmente o da representação clássica. Algoritmos quânticos baseados nestes princípios têm sido criados e testados, mostrando a validade da Computação Quântica e a possibilidade de redução da complexidade envolvida com determinados tipos de problemas (ROSS, 2008; BACON e LEUNG, 2007; AHARONOV, 1998). Talvez o caso mais emblemático tenha sido o algoritmo de fatoração de Shor. Este algoritmo reduz o problema da fatoração de números primos muito grandes a uma busca de ordem (SHOR, 1994; NIELSEN e CHUANG, 2005). Esta tarefa é considerada pelo critério da computação convencional como um problema intratável, e a utilização deste algoritmo quântico poderia reduzir drasticamente o tempo para decifrar chaves criptografadas. Sistemas de chaves de segurança públicas, utilizadas hoje em dia em criptografia, com base no algoritmo RSA de 256 bits carecem de tempo na ordem dos milhões de anos para serem decifradas. Entretanto, o algoritmo de Shor, utilizando a superposição e o emaranhamento, permitiria reduzir este tempo para a escala de horas (OLIVEIRA e LEITE, 2009). Este algoritmo já foi testado em computadores quânticos experimentais para a fatoração do número “15” (ROSS, 2008). Assim, com o desenvolvimento de modelos de computadores quânticos mais avançados e robustos, será possível obter benefícios em larga escala proporcionados por esta nova forma de computação.

Em face das situações apresentadas anteriormente e do grande potencial demonstrado pela Computação Quântica na resolução de problemas complexos, formula-se assim o problema de pesquisa ao qual se refere esta tese:

- É possível utilizar a Computação Quântica como solução para a representação e o processamento do conhecimento codificado em ontologias?

1.2 OBJETIVO GERAL

Propor um *framework* para análise de ontologias complexas, considerando os conceitos de **superposição** e **emaranhamento**, provenientes da Computação Quântica.

1.2.1 Objetivos Específicos

1) Identificar uma solução para a validação de instâncias de acordo com axiomas expressos em ontologias;

2) Identificar uma solução para evidenciar relações de transitividade entre classes de uma ontologia;

3) Identificar um procedimento para encontrar classes comuns em ontologias complexas para tarefas de alinhamento ou unificação de ontologias;

4) Desenvolver um simulador para Computação Quântica para suporte ao processo de criação dos algoritmos necessários ao processamento de ontologias complexas;

5) Elaborar exemplos e casos de aplicação das soluções em questão para ontologias complexas.

1.4 CONTEXTUALIZAÇÃO

O uso de ontologias como uma técnica para auxiliar a representação de conhecimento, na área da Gestão do Conhecimento, tem sido exaustivamente estudado. Ontologias se referem a uma conceituação consentida sobre uma determinada área de conhecimento (BORST, 1997; GÓMEZ-PÉREZ et al, 2004). Sua proposta é a de possibilitar a manipulação de significados ou semântica num sistema, indo além da representação sintática de dados apresentados em uma página da Web, ou ainda dados relacionados em um banco de dados ou bancos hierárquicos, por exemplo. A Web Semântica é uma alternativa para viabilizar este empreendimento, dotando a Web de um poder maior de representação de conhecimento (BERNERS-LEE et al, 2001). Portanto, sistemas ou agentes em portais podem manipulá-lo, e a descoberta de novas relações entre conhecimentos distintos pode ser alcançada.

Assim, as ontologias podem existir disseminadas pela Web, presentes nos portais, servindo de repositórios para que as relações entre conceitos entre sites distintos gerem novas descobertas, ou permitam relações que proporcionem resultados com significância. Softwares presentes em agentes inteligentes podem fazer validações nas

representações lá colocadas, permitindo a manutenção da consistência da ontologia (RUSSEL e NORVIG, 2004). Podem também capturar ontologias diferentes, e verificar conceitos que sejam comuns às representações. A partir destes relacionamentos, tais softwares podem informar aqueles que estejam acima de um limiar de significância, possibilitando ao usuário humano alcançar um novo conhecimento. Softwares em agentes inteligentes ainda podem ter outras funções, como proporcionar aquisição de novos conhecimentos, a partir de novos conceitos que são trabalhados pelo usuário e que podem, a partir de um dado grau de automatização, fazer parte das ontologias envolvidas (GÓMEZ-PÉREZ et al, 2004). Abordagens híbridas na tentativa de se conceber um modelo de memória de trabalho de curto e longo prazo para agentes, similar em estrutura ao modelo da mente humana e utilizando ontologias para representação de conceitos também foram propostas (MEDEIROS et al, 2006).

Entretanto, a agregação de novos conceitos em uma ontologia de um sistema isolado, ou o relacionamento entre várias ontologias em portais distintos exige, com o aumento crescente da rede de conceitos, a adoção de algoritmos eficientes e rápidos para o processamento. E este processamento tem complexidade crescente com o tamanho. Quando as ontologias possuem poucos conceitos, o processamento ocorre em tempo aceitável. Porém, conhecimentos realmente relevantes (ou seja, com similaridade àqueles originados da mente humana) tenderão a emergir em representações complexas de ontologias, e algoritmos simples, tais como o caminhamento transitivo, irão requerer alto custo para execução (TENENBAUM et al, 1995; CORMEN et al, 2002).

Outra questão pertinente é a abordagem algorítmica convencional das ontologias. Ainda que a representação de conceitos esteja em rede, permitindo múltiplas conexões, o método de raciocínio sobre a ontologia tende a ser linear, tanto para a criação quanto para a extração de informações. Ou seja, ontologias são **concebidas e inferidas de forma linear**. Mesmo com a possibilidade de paralelismo dos processos desta extração, o algoritmo é pensado de forma seqüencial, e cada parte deste processo acontece sequencialmente, em um ou mais processadores. Portanto, se o que se deseja com a representação das ontologias é obter vantagens pela aproximação da representação simbólica da mente humana, esta representação e os processos envolvidos na manipulação desta são parcos e não conseguem capturar a essência da complexidade envolvida, proveniente da multiplicidade de conexões. Assim, como resultado, obtém-se representações baseadas em

abstrações lineares e fundamentadas na causalidade local. Na direção desta insuficiência, Perseguers et al (2010) argumentam que uma rede de alta complexidade deve ser pensada de forma quântica, e isto implica em transcender a visão linear com a qual se lida em tais redes.

Em paralelo a isto, têm-se levantado uma série de hipóteses sobre o **argumento da insuficiência** do cérebro na produção dos processos da mente em termos unicamente das interações eletroquímicas entre neurônios (PENROSE, 1991; GOSWAMI et al, 1993; SATINOVER, 2008). Em cima deste argumento, surgiu a abordagem sobre a existência de fenômenos quânticos que, em hipótese, dariam suporte às atividades cerebrais e à mente humana. Certos mecanismos de pensamento apresentariam propriedades, tais como a superposição quântica e o emaranhamento, que poderiam explicar como o cérebro raciocina com muitas informações ao mesmo tempo de forma paralela, consciente e inconscientemente, num substrato material ou "hardware" que opera de certa forma lento. Alguns trabalhos têm sugerido paralelos entre efeitos quânticos e a compreensão de associações de palavras no pensamento humano (GOSWAMI et al, 1993; BRUZA et al, 2009).

Eccles (1990) também propôs uma teoria das interações entre as atividades mentais com os processos cerebrais onde haveria a presença de um campo de probabilidades quânticas nos processos de ativação dos neurônios, inserindo um grau de aleatoriedade de acordo com o princípio da incerteza de Heisenberg (EISBERG e RESNICK, 1979). Sua teoria se baseava em uma **proposta dualista** apresentada em conjunto com Popper, onde os eventos mentais tais como os pensamentos seriam considerados como elementos do mundo "2", estão separados do mundo dos processos cerebrais, o mundo "1" (POPPER e ECCLES, 1977; KAK, 1995).

Estudos envolvendo **esforços interdisciplinares** entre a área da Inteligência Artificial e da Computação Quântica e mesmo a Mecânica Quântica também tem sido amplamente propostos. Kak (1995) explorou o paralelo entre a computação neural quântica com o comportamento cerebral, apontando as limitações do paradigma computacional convencional, considerando este inapto para elaboração de esquemas envolvendo o processamento holístico, especulando inclusive que um computador neural quântico possa ser autoconsciente. Ezhov e Ventura (2000) elaboraram uma concepção de redes neurais quânticas argumentando que tais conceitos poderiam ajudar a elucidar o funcionamento do cérebro, bem como sugerir novos sistemas de informação, a solução de problemas intratáveis e memória associativa

com capacidade exponencial. Chrisley (1995) referiu-se às vantagens do aprendizado em algoritmos de redes neurais quânticas, que podem sobrepujar o poder de processamento dos atuais computadores.

Ainda de acordo com a perspectiva **conexionista** e incluindo também a evolucionária, vários modelos interdisciplinares têm sido propostos, incluindo: memória associativa quântica (VENTURA e MARTINEZ, 2000), aprendizado por reforço quântico (DONG et al, 2008), otimização genética quântica (MALOSSINI et al, 2008), design evolucionário de circuitos quânticos (REID, 2005; RUBINSTEIN, 2000), neurônios artificiais com propriedades quantum-mecânicas (VENTURA e MARTINEZ, 1997), controle fuzzy utilizando computação quântica (RIGATOS e TZAFESTAS, 2000), otimização quântica (HOGG e PORTNOV, 2000) e aprendizado de máquina e robótica utilizando propriedades da mecânica quântica (DONG et al, 2006a; DONG et al, 2006b).

Uma das razões mais relevantes que tem sido apresentada para justificar o desenvolvimento da Computação e Informação Quântica está relacionada à validade da **Lei de Moore**, formulada por Gordon Moore, um dos fundadores da Intel Corporation (MOORE, 1965). Elaborada ainda no início da era dos microprocessadores, de cunho mais empírico do que científico, ela previa que a indústria dobraria a velocidade de processamento dos processadores a cada 18 meses. Esta lei tem se mantido verdadeira e funcionado como um **padrão** para a indústria ao longo dos anos, porém tem se especulado o fim de sua vigência ainda para a segunda década deste século (NIELSEN e CHUANG, 2004). Os chips existentes no mercado trabalham com os transistores fabricados na escala de 45 e 32 nm (nanômetros, 1 nm=10^{-9}m), devendo ser lançados em 2010 chips já na faixa dos 28 nm (CULLIMORE, 2010). O limite consensuado para a continuidade da miniaturização dos transistores é de até 10 nm, sendo que abaixo desta escala, os efeitos quânticos impedem o funcionamento convencional dos transistores. Com a indústria já produzindo chips em escala nanométrica, vislumbra-se então o potencial para a construção de computadores quânticos. A Computação Quântica permite exceder os limites da computação convencional, a partir do uso de propriedades tais como a superposição de estados quânticos e o emaranhamento.

Tecnologias recentes, explorando propriedades de novos materiais tais como o grafeno[4], pressagiam alternativas de construção de

[4] Malha de átomos de carbono ligados entre si, com a largura de um átomo de espessura.

novos tipos ou ainda novas arquiteturas de microprocessadores, com freqüências de processamento podendo chegar até 1000 GHz, ou 1 THz[5] (WANG et al, 2009). Ainda que tais tecnologias venham a proporcionar uma 'sobrevida' à Lei de Moore, nos termos do padrão convencional, a viabilização da construção de computadores quânticos robustos em maior escala, esperada para este século (ROSS, 2007), tornarão as possibilidades de aplicação prática da Computação Quântica inevitavelmente maiores.

Portanto, a criação e extração de informações e conhecimento mediante ontologias baseadas neste paradigma quântico da computação, poderiam, em hipótese, serem potencializados de forma significativa. O uso do princípio da **superposição** permitiria lidar com as classes das ontologias assumindo os valores de todas as classes ou instâncias existentes ao mesmo tempo em um raciocínio ou inferência ontológica. E o uso do princípio do **emaranhamento** permitiria lidar com relações ocultas entre os conceitos presentes na ontologia, apresentando possibilidades de enorme ganho computacional para o processamento, e heurístico para a obtenção de novos conhecimentos.

Outro aspecto que vem a contribuir para a adoção do paradigma quântico em representação de ontologias se refere ao aspecto **probabilístico** inerente ao mundo quântico. Até que se faça a medição de um q-bit, este consegue "existir" em vários estados ao mesmo tempo, e a evolução deste q-bit no tempo modifica, de forma determinística, as probabilidades destes vários estados. Com a medida ou observação, apenas um dos estados se manifesta. Lidar com probabilidades está, portanto, de conformidade com a natureza dos processos de raciocínio envolvidos na mente humana (RUSSEL e NORVIG, 2004), e formas de raciocínio e inferência probabilística (assim como a existente em sistemas difusos) pode ter perfeitamente uma contrapartida quântica e serem utilizados em ontologias.

Por fim, ressalta-se que o relacionamento da teoria quântica com as técnicas de ontologias utilizadas neste trabalho está baseado no paradigma de Computação e Informação Quântica, que possui características tanto determinísticas quanto probabilísticas. E esta relação poderá se tornar bastante vantajosa para processos de Engenharia e Gestão do Conhecimento que venham a incluir a manipulação do conhecimento presente em ontologias complexas.

[5] Terahertz

1.5 MOTIVAÇÃO

A característica de ineditismo da tese está fundamentada principalmente na união das duas áreas de conhecimento, a Engenharia do Conhecimento (no que tange ao tópico Ontologias) e a Computação Quântica, **não tendo sido encontrado nas pesquisas bibliográficas nenhum trabalho neste sentido**. O trabalho de Bruza et al (2009) faz somente comparações entre a forma com que o ser humano pensa em situações onde parece haver uma associação de ideias tal como numa rede ou grafo, utilizando-se explicitamente do princípio do emaranhamento como metáfora para explicar as correlações que a mente faria de forma inconsciente. Porém, este trabalho não caracteriza o uso do paradigma quântico em ambiente computacional, mas buscando apenas explicar mecanismos da mente humana, na mesma linha dos trabalhos de John Eccles (1990), Roger Penrose (1991), Amit Goswami (1993) e Jeffrey Satinover (2007).

Outro aspecto relevante para consideração do ineditismo é a proposta de um modelo contendo **algoritmos quânticos** desenvolvidos para o processamento de ontologias, tais como a validação (quantificação existencial), o raciocínio transitivo e o processo de *merging* de ontologias. O modelo também avança no sentido de embutir na construção de ontologias os princípios de superposição e emaranhamento, resultando num poder de expressividade bem maior das mesmas.

O simulador de circuitos quânticos desenvolvido neste trabalho permitiu simulações experimentais da ordem de poucos q-bits, indicando o caminho para a construção dos algoritmos mais complexos que são descritos aqui. Elementos no design de circuitos quânticos, tais como as **memórias quânticas**, não possuem ainda na teoria da Computação Quântica uma representação específica (ainda que exista formalismo matemático explicando seu funcionamento), sendo sugerido aqui um formato de endereçamento-dado baseado na teoria de eletrônica digital convencional.

No estudo das implicações da teoria quântica, o termo "ontologia quântica" aparece no campo da filosofia relacionado à problemática da qualificação de objetos quânticos, conforme sua caracterização a partir do formalismo matemático da equação de Schröedinger, Relatividade Especial e outras teorias associadas (KRAUSE, 2000). Porém, é uma abordagem de cunho filosófico, com a preocupação em identificar uma espécie de meta-conhecimento que englobe a teoria quântica. Mesmo assim, evitou-se aqui a definição de um termo composto "ontologia

quântica", para não haver sobreposição com tal conceituação filosófica. Uma ontologia quântica pode considerar, por exemplo, a existência real tanto da onda quanto da partícula, como no caso da mecânica quântica de Bohm-de Broglie (KNOLL, 2006). Entretanto, considerando-se a interpretação de Copenhaguen, onda e partícula são as manifestações complementares exclusivas de um único objeto quântico (PESSOA JUNIOR, 2003).

1.6 ESCOPO DO TRABALHO

O desenvolvimento desta tese esteve focado no uso da Computação Quântica como ferramenta para a área da Engenharia do Conhecimento, restrita ao âmbito das ontologias. Não faz parte da tese qualquer incursão ou estudo das conseqüências advindas disto para a área neurocognitiva. O foco no desenvolvimento do modelo está centrado no processamento e construção de ontologias, com alguns algoritmos sendo apresentados neste sentido para demonstrar a validade do argumento.

Na consideração do tema da Computação Quântica, é feita somente a abordagem que contribui para o desenvolvimento dos algoritmos, e neste sentido a área da Informação Quântica não é explorada em termos, por exemplo, de algoritmos de correção quântica de erros. A apresentação da Computação Quântica é restrita aos aspectos de computação começando pelo modelo já consolidado do *q-bit* e evitando-se a abordagem extensiva da Mecânica Quântica. Ressalta-se também que o desenvolvimento dos algoritmos quânticos seguiu uma linha genérica, não se atendo a algum paradigma específico de hardware para Computação Quântica.

Os algoritmos de processamento apresentados possuem ao final um estudo preliminar da complexidade envolvida na resolução dos mesmos, baseada predominantemente na vantagem dos algoritmos ou sub-rotinas quânticas em relação aos clássicos. Mas deve-se ressaltar que este estudo é relativo, devendo-se considerar que os diferentes hardwares quânticos podem requerer tempos de operações diferenciados, e relações de compromisso devem ser buscadas para estudar a vantagem efetiva.

Portanto, o presente trabalho está focado na análise de ontologias complexas já existentes, construídas a partir de diversas metodologias presentes na literatura sobre engenharia ontológica. Não serão tratados métodos referentes à construção de novos tipos de ontologias.

1.7 ASPECTOS METODOLÓGICOS

A metodologia do trabalho é explicada sob a perspectiva da pirâmide metodológica de Schreiber et al (2002), adaptada para a tese e sendo visualizada na Figura 2. Na camada referente à **visão de mundo**, a fundamentação para a tese começa na adoção de uma visão complexa do mundo, considerando-se as várias inter-relações existentes entre as coisas e eventos, e procurando-se expandir a visão clássica da realidade para um paradigma quântico, como já mencionado anteriormente na contextualização.

Na camada referente à **teoria**, contribui o corpo de conhecimentos referente à área de Ontologias, que fazem parte da Engenharia do Conhecimento, e a área da Computação Quântica. O campo comum a tais áreas (atribuindo-se o nome de "Ontologias-CQ") contém os elementos relativos à construção e processamento de Ontologias, utilizando os princípios da superposição e emaranhamento provenientes da Computação Quântica em vários aspectos de Ontologias, tais como classes, instâncias e relações.

Na camada de **métodos** posiciona-se o modelo proposto, consistindo assim num *framework* que abrange o estudo e desenvolvimento de quatro subáreas derivadas: superposição de classes, superposição de instâncias, superposição de relações e emaranhamento de classes. Em cada subárea derivada, são apresentados e explicados os algoritmos, alguns de maneira mais formal e esquemática.

Na camada de **ferramentas**, os instrumentos que estão à disposição para a aplicação do modelo são: (i) o formalismo matemático, que inclui o uso de conceitos da álgebra linear; e (ii) programas simuladores de Computação Quântica. Na atualidade, ainda não se dispõe de computadores quânticos robustos para processar a quantidade de q-bits exigida pelos algoritmos, e isso justifica a importância para uso de métodos de **simulação** (DODIG-CRNKOVIC, 2002).

Por fim, na camada de **uso**, os algoritmos presentes no *framework* são explicados na forma de *cases* de ontologias, auxiliando na visualização prática de uso do modelo que podem, por sua vez, serem utilizados em sistemas de conhecimento para suporte a processos de Gestão do Conhecimento.

O desenvolvimento dos cases permitiu retroagir tanto para o melhoramento das ferramentas (no caso, o software de simulação) quanto para o refinamento dos algoritmos propostos no framework. O simulador sofreu algumas adaptações, tanto para comportar a

representação de memória quântica quanto para a execução dos algoritmos no escopo de q-bits possível do software, e também para permitir o design de circuitos quânticos em maior escala de q-bits.

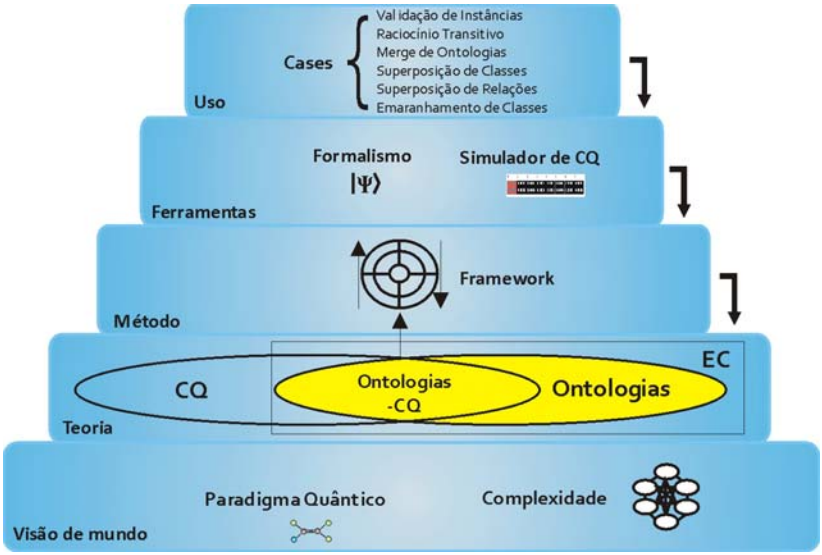


Figura 1: Metodologia da tese de acordo com a pirâmide metodológica de Schreiber et al (2002).
Fonte: Elaborado pelo autor.

1.8 ADERÊNCIA AO PROGRAMA DE ENGENHARIA E GESTÃO DO CONHECIMENTO

Ontologias são largamente usadas em Engenharia do Conhecimento, em aplicações relacionadas à Gestão do Conhecimento (GÓMEZ-PÉREZ et al, 2004). As ontologias também são consideradas como uma das maiores realizações da Engenharia do Conhecimento, ao lado dos métodos para resolução de problemas (STUDER et al, 2000).

Sistemas de Gestão do Conhecimento devem estar aptos para a geração, descoberta, captura, retenção, distribuição e aplicação de uma grande variedade de conhecimento explícito, fornecido através de serviços baseados em conhecimento. Além disso, um sistema de Gestão do Conhecimento necessita de capacidades para interoperabilidade com outros sistemas organizacionais. Para satisfazer estes requisitos, um sistema de Gestão do Conhecimento precisa de capacidade de

representação de conhecimento, e isto pode ser conseguido através das ontologias (GUALTIERI e RUFFOLO, 2005).

O presente trabalho tem foco na análise de ontologias complexas, mediante a proposição de algoritmos que permitam a extração eficiente de conhecimento relevante, em domínios de complexidade crescente. Isto é alcançado com a contribuição interdisciplinar proveniente da área da Computação Quântica (NIELSEN e CHUANG, 2004), ampliando o escopo de atuação da Engenharia do Conhecimento através dos conceitos de superposição e emaranhamento.

Mesmo que não existam ainda computadores quânticos robustos, um leque de alternativas se abre com a união destas áreas, evidenciando-se o potencial de aplicação em cenários onde a tendência é o aumento crescente da representação de conhecimento em ontologias, para suporte aos processos de Engenharia e Gestão do Conhecimento.

2 FUNDAMENTAÇÃO TEÓRICA

O modelo geral a ser apresentado faz interface com as áreas da Engenharia do Conhecimento (Ontologias) e a Computação Quântica, considerados como os dois pilares do modelo. Assim, a fundamentação teórica deste trabalho está dividida em duas partes: uma referente aos conceitos de Engenharia Ontológica, e outra referente à Computação Quântica.

2.1 ONTOLOGIAS

Nesta seção são apresentados os conceitos de ontologias, componentes e tipos de componentes, construção e modelagem, categorização de ontologias, princípios para a construção, linguagens, relação das ontologias com a Engenharia do Conhecimento e mecanismos de inferência de ontologias. Visa-se, com isso, fornecer a base para a elaboração dos constituintes do *framework* a ser caracterizada posteriormente de forma incremental.

2.1.1 Conceitos de Ontologias

As ontologias têm como ponto de partida a Filosofia e o estudo da essência das coisas, remontando a Parmênides de Eléia (530? a.C.-460? a.C), a quem primeiro se associa a ideia da separação entre a essência das coisas com aquilo que é percebido pelo ser humano; e a Aristóteles (384 a.C.-322 a.C), que introduziu a ideia de "potência" e "ação das coisas" e estabeleceu um sistema de categorias para distinguir os diferentes modos de "ser" para classificar as coisas (GÓMEZ-PÉREZ et al, 2004). Emmanuel Kant (1724-1804) colocou que a essência das coisas não é determinada pelas coisas em si, mas também pela contribuição de quem as percebe e compreende. Ortega y Gasset (1883-1955) foi mais além e argumentou que o mundo é fortemente dependente da maneira como as pessoas percebem-no. E Williams James (1842-1910) considerou a verdade como aquilo que cada pessoa considera como tendo as melhores conseqüências. Gómez-Pérez et al (2004) assinalam que os Sistemas de Informação seguem tanto a visão de Ortega y Gasset quanto de James, pois as estruturas de dados e as bases de conhecimento são modeladas não apenas para representar o mundo, mas também para se trabalhar de forma mais eficiente, de acordo com a proposta pela qual tal sistema foi modelado. Desta forma, as ontologias têm-se evidenciado como um importante recurso na área

da Ciência da Computação e mostrando grande similaridade com o conceito de redes semânticas (cujo início é associado a Pierce, ainda em 1909) e o conceito de *frames* introduzido por McCarthy e Hayes, que fazem parte da linha simbólica na área de Inteligência Artificial (RUSSEL e NORVIG, 2004, p.25, p.309).

Ao longo do desenvolvimento da teoria sobre ontologias, uma série de conceitos foi elaborada por vários autores, como mostra o Quadro 1. A partir deste quadro, fica evidente que uma ontologia consiste num grupo de termos ou conceitos referentes a uma área de conhecimento, que estão relacionados entre si, sendo um consenso em termos de compreensão para um grupo de pessoas que a compartilham de forma comum.

Autor(es)	Conceito
Neches et al (1991)	Uma ontologia define os termos básicos e relações compreendendo o vocabulário de uma área assim como as regras para combinar termos e relações para definir extensões para o vocabulário
Gruber (1993)	Uma ontologia é uma especificação explícita de uma conceituação
Borst (1997)	Ontologias são definidas como uma especificação formal de uma conceituação compartilhada
Studer et al (1998)	Uma ontologia é uma especificação formal e explícita de uma conceituação compartilhada
Guarino e Giaretta (1995)	Uma teoria lógica que dá uma consideração parcial e explícita de uma conceituação
Guarino (1998)	Um conjunto de axiomas lógicos desenhados para endereçar o significado pretendido de um vocabulário
Schreiber et al (1995)	Uma ontologia fornece os meios para descrever explicitamente a conceituação por trás do conhecimento representado em uma base de conhecimento
Uschold e Gruninger (1996)	Termo utilizado para se referir a uma compreensão compartilhada de algum domínio de interesse, o qual deve ser

	usado como um framework unificador para resolver problemas de comunicação, interoperabilidade e reutilização.
Swartout et al (1997)	Uma ontologia é um conjunto de termos hierarquicamente estruturados para descrever um domínio que pode ser usado como "esqueleto" para uma base de conhecimento
Uschold e Jasper (1999)	Assume uma variedade de formas, mas isto irá necessariamente incluir um vocabulário de termos e alguma especificação de seu significado. Inclui definições e uma indicação de como conceitos estão inter-relacionados cuja coletividade impõe uma estrutura no domínio e restringe as interpretações possíveis dos temas
Gruber (2008)	Uma ontologia especifica um vocabulário com o qual se faz asserções que podem ser entradas ou saídas de agentes inteligentes, tais como um programa de software

Quadro 1: Diferentes conceitos de ontologia
Fonte: Elaborado pelo autor.

Studer et al (1998), no detalhamento de seu conceito, explicam:

1) **Conceituação,** se referindo a um modelo abstrato de algum fenômeno no mundo como tendo identificado os conceitos relevantes do fenômeno;

2) **Explícito,** significando que os tipos de conceitos usados e as restrições sobre eles são explicitamente definidos;

3) **Formal,** se referindo ao fato de que a ontologia deveria ser lida de forma automática;

4 **Compartilhada,** refletindo a noção de que uma ontologia captura conhecimento consensual, isto é, não é próprio de algum indivíduo, porém aceito por um grupo. Esta questão do consenso é aprofundada em tópico específico mais adiante.

Guarino e Giaretta (1995) coletaram e analisaram as seguintes definições de ontologias como:

1)Um **sistema filosófico**;
2)Um **sistema conceitual informal**;

3)Uma **conta semântica formal**;
4)Uma **conceituação** de uma **especificação**;
5)Uma **representação** de um **sistema conceitual** mediante teoria lógica;
6)Um **vocabulário** utilizado por uma **teoria lógica**;
7)Uma **especificação** em **meta-nível** de uma **teoria lógica**.

Welty et al (1999), usando uma definição estruturada como a de Guarino e Giaretta, consideram que uma ontologia varia num *continuum* de complexidade, relacionando ainda ao uso de raciocínio automatizado. Na medida em que evolui este *continuum*, a ontologia pode ser:

1)Um **catálogo**;
2)Um conjunto de **arquivos de texto**;
3)Um **glossário**;
4)Um **tesauro** (léxico);
5)Uma coleção de **taxonomias**;
6)Uma coleção de **frames**;
7)Um conjunto de **restrições lógicas gerais**.

Numa definição mais recente, Gruber (2008) afirma que uma ontologia especifica um vocabulário com o qual se faz asserções que podem ser entradas ou saídas de agentes inteligentes, tais como um programa de software. Como uma especificação de interface, a ontologia fornece uma linguagem para comunicação com o agente. Não há necessidade do uso dos termos da ontologia como uma codificação interna do seu conhecimento. As definições e restrições formais da ontologia colocam limitações no que pode ser significativo representado nesta linguagem. Em essência, comprometer-se com uma ontologia (ou seja, manter uma interface utilizando o vocabulário da ontologia) requer que declarações que são afirmadas sobre entradas e saídas sejam logicamente consistentes com as definições e restrições da ontologia.

Como exemplos práticos de ontologias, podem-se citar:

1)**Protégé Ontology Library** (NOY et al, 2000);
2)**Cyc** (REED e LENAT, 2002);
3)**WordNet** (GOMEZ-PEREZ et al, 2004);
4)**DAML** (HORROCKS e VAN HARMELEN, 2001).

2.1.2 Componentes Principais de uma Ontologia

Conforme o formalismo em que foram expressas as ontologias, elas podem ser (GOMEZ-PEREZ et al, 2004):

1)**Altamente informais**, caso sejam expressas em linguagem natural;
2)**Semi-informais**, se expressas em uma estrutura restrita de uma linguagem natural;
3)**Semi-formais**, se expressas em uma linguagem definida formalmente e de uso artificial, tais como OWL (DEAN e SCHREIBER, 2003);
4)**Rigorosamente formais**, se elas fornecem meticulosamente termos definidos com semântica formal, teoremas e provas de propriedades tais como validade e completeza.

No início dos anos 90, ontologias eram construídas utilizando-se principalmente técnicas de modelagem de IA baseadas em frames e lógica de primeira ordem. Nos últimos anos, outras técnicas de representação de conhecimento baseadas em descrição lógica (BAADER et al, 2003) tem sido utilizadas em linguagens tais como OIL (HORROCKS et al, 2000), DAML+OIL (HORROCKS e VAN HARMELEN, 2001) e OWL (DEAN e SCHREIBER, 2003).

Existem importantes conexões e implicações entre o conhecimento modelando **componentes** (conceitos, regras, etc.), o conhecimento representando **paradigmas** (*frames*, descrição lógica, lógica) utilizados para representar formalmente tais componentes; e as **linguagens** utilizadas para implementar as ontologias sob certo paradigma de representação de conhecimento.

Gómez-Perez et al (2004) diferenciam ontologias entre *lightweight* (“peso-leve”) e *heavyweight* (“peso-pesado”), ressaltando que as primeiras podem ser construídas a partir de abordagens baseadas em engenharia de software (UML) e bancos de dados (Diagramas de Entidade-Relacionamento). Entretanto, as ontologias *heavyweight* necessitam de abordagens baseadas em técnicas de IA para combinar frames e a lógica de primeira ordem.

Outras técnicas utilizadas largamente em engenharia de software ou bancos de dados para modelagem de conceitos, relacionamentos entre conceitos e atributos de conceitos poderiam ser apropriadamente utilizados para construir ontologias *lightweight*, devido a estas técnicas já imporem uma estruturação sobre o domínio de conhecimento e restringe a interpretação dos termos. Porém, é importante ressaltar que o modelo pode apenas ser considerado como uma ontologia se o modelo

de conhecimento consensual compartilhado é aceito pela comunidade de interesse (GOMEZ-PEREZ et al, 2004).

2.1.3 Tipos de Componentes de uma Ontologia

Os tipos de componentes de uma ontologia variam de acordo com o uso de frames e lógica de primeira, ou o uso de frames e lógica descritiva. Gruber (1993) propôs a modelagem de ontologias utilizando frames e lógica de primeira ordem, identificando cinco tipos de componentes (GOMEZ-PEREZ et al, 2004):

1)**Classes**: representam conceitos os quais são utilizados em um amplo sentido. As classes em uma ontologia são usualmente organizadas em taxonomias através de um mecanismo de herança. As classes ainda podem representar conceitos abstratos (intenções, crenças, sentimentos, etc.) ou conceitos concretos (pessoas, computadores, coisas, etc.)

2)**Relações**: representam um tipo de associação entre os conceitos de um domínio. Elas são formalmente definidas como um subconjunto de um produto de *N* conjuntos $R \subset C_1 \times C_2 \times ... C_N$. As ontologias normalmente utilizam relações binárias.

3)**Funções**: consideradas como um caso especial de relações nas quais o *n*-ésimo elemento da relação é único para os *n-1* elementos que o precedem. Funções podem ser expressas formalmente como $F : C_1 \times C_2 \times ... C_{N-1} \rightarrow C_N$.

4)**Axiomas**: servem para modelagem de sentenças tautológicas, ou seja, que são sempre verdadeiras. São normalmente usadas para representar conhecimento que não pode ser formalmente definido por outros componentes. Em adição a isto, axiomas formais podem ser utilizados para verificar a consistência da própria ontologia ou a consistência do conhecimento armazenado numa base de conhecimento. Os axiomas formais também são muito úteis para inferência de novos conhecimentos. Eles ainda podem ser independentes de domínio.

5)**Instâncias**: são utilizadas para representar elementos ou indivíduos em uma ontologia, podendo-se instanciar conceitos ou relações.

Outra tipificação de componentes baseada em frames é mostrada por Noy et al (2000) com referência ao modelo de conhecimento da plataforma *Protégé*-2000:

1) **Classes**: são conceitos no domínio de discurso;
2) **Slots**: descrevem atributos ou propriedades das classes;
3) **Facets**: descrevem atributos ou propriedades dos slots;
4) **Axiomas**: especificam restrições adicionais;
5) **Instâncias**: indivíduos pertencentes às classes.

A Lógica Descritiva (*Description Logics* - DL) é um formalismo lógico utilizado para descrever ontologias *heavyweight*, onde algumas implementações em termos de linguagens e sistemas foram a LOOM (MacGregor, 1991), e Kris (Baader e Hollunder, 1991).

De acordo com Baader et al (2003), uma teoria de DL se divide em duas partes: i) *TBox*: contém o conhecimento na forma de terminologia e é construído através de declarações que descrevem as propriedades gerais dos conceitos. Seriam os conceitos e regras; e ii) *ABox*: contém o conhecimento afirmativo, o que é específico para os indivíduos do domínio do discurso, ou seja, as instâncias.

Os tipos de componentes de uma ontologia sob a ótica da DL se diferenciam ligeiramente daqueles apontados por Gruber (1993):

1)**Conceitos**: tem o mesmo significado no paradigma de *frames*, consistindo nas classes de objetos. Os conceitos em DL podem ser primitivos ou definidos.
2)**Regras**: descrevem as relações binárias entre conceitos, permitindo também a descrição de propriedades dos conceitos. Relações de ordem superior também são permitidas em algumas linguagens de DL. As regras podem ser primitivas ou derivadas.
3)**Indivíduos**: representam as instâncias das classes. São as instâncias de conceitos e os valores de suas regras (ou seja, propriedades). Os sistemas de DL normalmente separam os indivíduos das descrições de conceitos e regras.

2.1.4 Categorização de Ontologias

Devido à multiplicidade de formas com que se pode montar uma ontologia, a divisão das ontologias em categorias foi uma das preocupações de vários autores. Gómez-Perez (2004) resume as classificações dos tipos de ontologias de vários autores de acordo com o objeto de contextualização, apresentados no Quadro 2.

Tipo	Descrição	Autor(es)
Ontologias de Representação de Conhecimento	Capturam as primitivas de representação utilizadas para formalizar conhecimento de acordo com determinado paradigma	Van Heijst et al (1997)
Ontologias Gerais ou Comuns	São utilizadas para representar conhecimento de senso comum reutilizável em vários domínios, incluindo vocabulário relacionado a coisas, eventos, tempo, espaço, causalidade, comportamento, função, etc.	Van Heijst et al (1997) Mizoguchi et al (1995)
Ontologias de Alto Nível	Descrevem conceitos muito generalizados e fornece as noções sob as quais todos os termos raiz nas ontologias existentes deverão estar ligados.	Guarino (1998)
Ontologias de Domínio	São ontologias reusáveis para um domínio específico em questão, fornecendo vocabulários sobre conceitos pertencentes a um domínio e seus relacionamentos, sobre as atividades que tomam lugar neste domínio e sobre as teorias e princípios elementares que governam este	Van Heijst et al (1997) Mizoguchi et al (1995)

	domínio.	
Ontologias de Tarefa	Descrevem o vocabulário relacionado a uma tarefa genérica ou atividade, tais como um diagnóstico ou agendamento, mediante a especialização de termos genéricos em ontologias de alto nível.	Mizoguchi et al (1995) Guarino (1998)
Ontologias de Domínio-Tarefa	São ontologias reutilizáveis em um domínio específico, mas não entre domínios caracterizando-se, entretanto, como indepedentes de aplicações.	Gómez-Pérez et al (2004)
Ontologias de Metodologia	Fornecem definições dos conceitos relevantes e relações aplicadas para especificar um processo de raciocínio ou ainda executar uma tarefa em particular.	Tijerino e Mizoguchi (1993)
Ontologias de Aplicação	São as ontologias dependentes de aplicações, contendo todas as definições necessárias para modelar o conhecimento requerido para uma aplicação em particular.	Van Heijst et al (1997)

Quadro 2: Categorias de ontologias.
Fonte: Adaptado de GÓMEZ-PÉREZ et al (2004).

2.2 CONSTRUÇÃO DE ONTOLOGIAS

A construção e modelagem de ontologias consistem em tarefas que consomem bastante tempo dos desenvolvedores. Tais tarefas se tornam mais complexas quando se busca implementar uma linguagem de ontologia sem uma ferramenta adequada de suporte. Várias

ferramentas para se trabalhar com ontologias em diversas atividades foram concebidas. Gómez-Pérez e Corcho (2002) diferenciam as ferramentas nos seguintes grupos:

1) **Desenvolvimento de Ontologias**: incluem ferramentas e suítes integradas para criação de novas ontologias, edição textual e gráfica, navegação, documentação, exportação e importação em diferentes formatos e linguagens e gestão de bibliotecas de ontologias;
2) **Avaliação de Ontologias**: ferramentas utilizadas para avaliar o conteúdo de ontologias e suas tecnologias relacionadas, para tentar reduzir problemas de integração e uso de ontologias em outros sistemas de informação;
3) **Merging e Alinhamento de Ontologias**: utilizadas para resolver o problema de unificar ontologias relativas a um mesmo domínio;
4) **Anotação de Ontologias**: são ferramentas utilizadas por usuários das ontologias para inserir instâncias de conceitos e relações em ontologias e manter, de forma automática ou semi-automática, marcações baseadas em ontologias em páginas da Web. Tais ferramentas são aproveitadas no contexto da Web Semântica;
5) **Consulta e Inferência de Ontologias**: tais ferramentas permitem consultas em ontologias e raciocínio por inferência, sendo fortemente relacionados à linguagem na qual a ontologia é implementada.
6) **Aprendizado de Ontologias**: ferramentas que podem direcionar a construção de ontologias de forma automática ou semi-automática a partir de formatos mais básicos de representação de conhecimento como textos em linguagem natural e bancos de dados.

Alguns exemplos de plataformas para modelagem de ontologias, que implementam várias destas atividades ou ferramentas, são:

1)**Protégé-2000** (NOY et al, 2000) ;
2)**WebODE** (ARPÍREZ et al, 2003; CORCHO et al, 2002);
3)**OpenCyc** (REED e LENAT, 2002);
4)**OntoKEM** (RAUTENBERG et al, 2009).

2.2.1 Princípios de Construção de Ontologias

A construção de ontologias deve respeitar alguns critérios para se garantir no final a obtenção de uma ontologia que represente o conhecimento alvo da forma mais coerente possível. Gruber (1993b) estabelece um conjunto de 5 (cinco) princípios para servirem de guia na elaboração e ainda avaliação de uma ontologia, e em Arpírez et al (1998) são dispostos mais dois critérios e um sugerido por Gómez-Pérez (2004), conforme o Quadro 3.

Princípio	Descrição	Autor(es)
Clareza	"Uma ontologia deve comunicar efetivamente o significado pretendido de termos definidos. Definições devem ser objetivas. Definições podem ser declaradas em axiomas formais, e uma completa definição (caracterizada por condições necessárias e suficientes) é preferida sobre uma definição parcial (caracterizada apenas por condições necessárias ou suficientes). Todas as definições devem ser documentadas em linguagem natural."	Gruber (1993b)
Viés Mínimo de Codificação	"A conceituação deve ser especificada ao nível de conhecimento sem depender de uma codificação em nível simbólico particular."	Gruber (1993b)
Extensibilidade	"Deve-se estar apto para a definição de novos termos para usos especiais baseados em vocabulário existente, em um modo que não venha a requerer a revisão das definições existentes."	Gruber (1993b)
Comprometimento Mínimo	"Desde que o comprometimento ontológico é baseado no uso consistente do vocabulário, tal comprometimento deve ser minimizado pela especificação da teoria mais fraca e definindo apenas aqueles termos que são essenciais para a comunicação de conhecimento consistente com a teoria."	Gruber (1993b)

Coerência	"Uma ontologia deve ser coerente, isto é, deve aprovar inferências que são consistentes com as definições (...) Se uma sentença que pode ser inferida dos axiomas contradiz a definição ou um exemplo dado informalmente, então a ontologia é inconsistente."	Gruber (1993b)
Representação de Conhecimento Disjuntivo e Completo	"Se o conjunto de subclasses de um conceito é disjunto, pode-se definir uma decomposição disjunta."	Gómez-Pérez et al (2004)
Minimização da Distância Sintática entre Conceitos Similares	"Conceitos similares são agrupados e representados como subclasses de uma classe e devem ser definidos utilizando-se as mesmas primitivas, onde os conceitos que são menos similares ficam representados mais além na hierarquia."	Arpírez et al (1998)
Padronização de Nomes	"(...) uma relação deve ser nomeada concatenando-se o nome da ontologia (ou o conceito representando o primeiro elemento da relação), o nome da relação e o nome do conceito-alvo."	Arpírez et al (1998)

Quadro 3: Princípios de construção de ontologias
Fonte: Elaborado pelo autor.

2.2.2 Metodologias para Construção de Ontologias

Até meados da década de 1990, a construção de ontologias era mais caracterizada como arte do que uma atividade de engenharia. Várias metodologias foram propostas para tornar a construção de ontologias um processo bem definido, seja através de um conjunto de princípios e critérios de design ou ainda por meio de fases para uma construção manual de uma ontologia (GÓMEZ-PÉREZ et al, 2004). Várias metodologias têm sido propostas desde então, sendo algumas mostradas no Quadro 4.

Tipo	Descrição dos Processos Principais	Autor(es)
Metodologia Cyc	1) Codificação manual de artigos e pedaços de conhecimento; 2) Codificação do conhecimento **auxiliado** por ferramentas, utilizando o conhecimento já armazenado na base de conhecimento; 3) Codificação de conhecimento **executada** principalmente por ferramentas, utilizando o conhecimento já armazenado na base de conhecimento.	Lenat e Guha (1990)
Metodologia de Uschold e King	1) Identificação do propósito; 2) Construção da ontologia – mediante a captura, codificação e integração de ontologias existentes; 3) Avaliação; 4) Documentação.	Uschold e King (1995)
Metodologia de Grüninger e Fox	1) Identificação de cenários motivadores; 2) Elaboração informal de questões de competência; 3) Especificação da terminologia utilizando a lógica formal; 4) Descrever as questões de competência de maneira formal, utilizando terminologia formal; 5) Especificar axiomas utilizando a lógica de primeira ordem; 6) Especificar teoremas de completeza.	Grüninger e Fox (1995)

Abordagem KACTUS	1) Especificação da aplicação; 2) Design preliminar baseado em categorias ontológicas de alto nível relevantes; 3) Refinamento da ontologia e estruturação.	Schreiber et al (1995)
METHONTOLOGY	1) Atividades de **Gerenciamento** 1.1) Agendamento 1.2) Controle 1.3) Garantia da Qualidade 2) Atividades de **Desenvolvimento** 2.1) Especificação 2.2) **Conceptualização** 2.3) **Formalização** 2.4) Implementação 2.5) Manutenção 3) Atividades de **Suporte** 3.1) Aquisição de conhecimento 3.2) Integração 3.3) Avaliação 3.4) Documentação 3.5) Gerenciamento de Configuração	Fernández-López et al (1997) Gómez-Pérez (1998)
Metodologia Baseada no SENSUS	1) Identificar termos "sementes" (conceitos-chave do domínio); 2) Ligar manualmente os termos "sementes" ao SENSUS; 3) Adicionar trilhas ao raiz do SENSUS; 4) Adicionar novos termos ao domínio; 5) Adicionar sub-árvores completas.	Swartout et al (1997)
Metodologia On-To-Knowledge	1) Estudo de factibilidade; 2) "Pontapé inicial" da ontologia; 3) Refinamento; 4) Avaliação; 5) Manutenção.	Staab et al (2001)

Quadro 4: Metodologias para construção de ontologias.
Fonte: Adaptado de Gómez-Pérez et al (2004).

Gómez-Pérez et al (2004) apresentam ainda um comparativo entre as várias metodologias mostrando as características de abordagem de cada uma. Na linha referente à METHONTOLOGY, estão grifadas as fases de conceptualização e formalização, pois serão abordadas na discussão da seção 2.3.3, referente ao tema do consenso.

2.2.3 Linguagens de Ontologias

A representação do conhecimento através de ontologias é feita mediante a escolha de uma linguagem ou um conjunto de linguagens nas quais tais ontologias serão implementadas de maneira formal. Várias linguagens para ontologias foram então concebidas. Geralmente a seleção de uma linguagem de ontologia não é baseada no modelo de representação de conhecimento ou nos mecanismos de inferência necessários pela aplicação que utilizará a ontologia, porém centrada nas preferências individuais dos desenvolvedores (GÓMEZ-PÉREZ et al, 2004).

Podem-se dividir as linguagens de ontologias em dois tipos:

1) **Linguagens Tradicionais**: as linguagens tradicionais foram criadas no começo da década de 1990, sendo baseadas em lógica de primeira ordem ou combinando com frames e lógica descritiva. A Figura 2 mostra um esquema representando as linguagens tradicionais. Um estudo comparativo demonstrou que LOOM e Ontolingua são as linguagens que oferecem mais recursos, permitindo alta representatividade de uma ontologia.

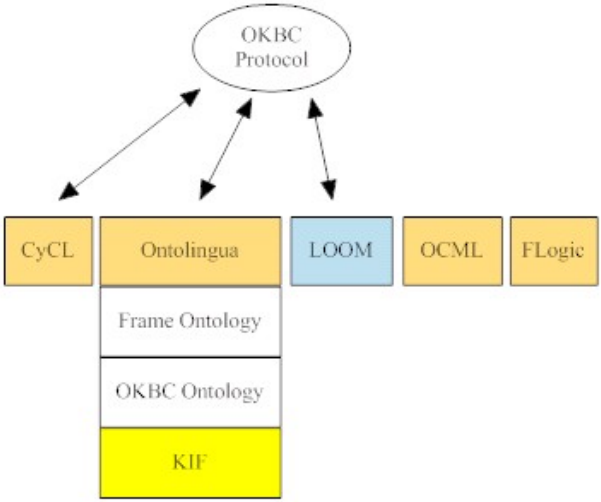


Figura 2: Linguagens tradicionais de ontologias.
Fonte: Adaptado de Gómez-Pérez (2004).

2) **Linguagens de Marcação**: são linguagens de ontologias que exploram as características da Web, sendo a sintaxe baseada em linguagem HTML ou XML. Na Figura 3 está descrito as principais linguagens de ontologias baseadas na Web. Uma das linguagens de marcação mais utilizadas atualmente é a OWL (DEAN e SCHREIBER, 2003), em razão de ser considerada uma evolução das linguagens OIL e DAML+OIL.

A escolha de uma linguagem pode ser determinante para a forma como a ontologia será representada. Algumas linguagens permitem a construção de ontologias "peso-pesado", com axiomas, funções e regras, enquanto outras (geralmente as linguagens de marcação) permitem apenas a representação de ontologias "peso-leve". Assim, existem diferenças relevantes entre a expressividade que uma ontologia pode ter, utilizando-se ou linguagens tradicionais ou linguagens de marcação (GÓMEZ-PÉREZ et al, 2004).

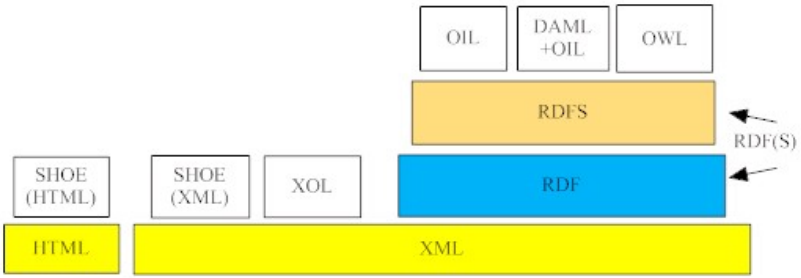


Figura 3: Linguagens de marcação de ontologias.
Fonte: Adaptado de Gómez-Pérez et al (2004).

2.3 ONTOLOGIAS E ENGENHARIA DO CONHECIMENTO

Studer et al (2000) comentam que ontologias tem sido um tópico bastante investigado pela comunidade acadêmica, incluindo pesquisadores de IA, engenharia do conhecimento, processamento de linguagem natural e representação do conhecimento. Ontologias encontram grande potencial de utilização na compreensão comum e compartilhada de um domínio que pode ser comunicado através das pessoas e sistemas.

Dependendo do seu nível de generalidade, diferentes tipos de ontologias podem ser identificados, os quais preenchem certas necessidades em processos de construção de sistemas baseados em conhecimento:

- **Ontologias de domínio**: capturam o conhecimento válido para um domínio particular, por exemplo, eletrônico, médico, mecânico, etc.
- **Ontologias genéricas ou de senso comum**: Capturam o conhecimento geral sobre o mundo e fornecem noções básicas e conceitos para coisas tais como tempo, espaço, estados, eventos, etc.
- **Ontologias de representação**: não estão comprometidas com um domínio em particular. Tais ontologias fornecem entidades representacionais, sem as quais os conceitos não poderiam ser representados. A meta-informação sobre determinado domínio pode ser construída em ontologias de representação.

Studer et al (2000) também ressaltam que parte da comunidade de pesquisa em ontologias está interessada na visualização e construção de tecnologias que habilitem o **reuso** em larga escala de ontologias em um nível bastante abrangente. Na medida em que tal reuso seja possível, as ontologias devem ser construídas em pequenos módulos com alta coerência interna e uma quantidade de interações limitadas entre os módulos.

2.3.1 Métodos Básicos e Técnicas

O acesso, a busca e a consolidação das informações permanecem como uma tarefa difícil, dado a quantidade massiva de informações existentes nas fontes de dados, particularmente na Web. As razões são as mais variadas, porém a mais significativa se refere à grande lacuna existente entre a concepção da informação tal como é vista pelo usuário e aquela que é armazenada e fornecida pelos sistemas de informação (STUDER et al, 2000). A grande questão reside, portanto, no estabelecimento de uma ponte para preencher esta lacuna e qual direção que minimiza o esforço de engenharia para um grande número e muitas variedades de fontes de informação. Isto inclui textos livres, informação semi-estruturada (XML, por exemplo) e informação em bancos de dados que exibem problemas similares de contextualização comum (Figura 4).

Ontologias de domínio podem ser uma solução parcial para tais problemas de integração. Além da própria integração, troca e extração, as ontologias de domínio podem permitir a descrição precisa de uma conceituação comum para várias fontes de informação.

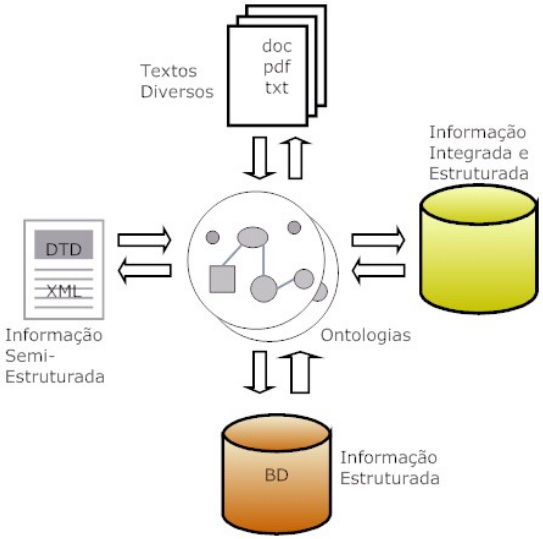


Figura 4: Integração de informação e extração de fontes de dados diversas usando ontologias.
Fonte: Adaptado de Studer et al (2000).

Studer et al (2000) ressaltam também as áreas de aplicação da Engenharia do Conhecimento e uma variedade de trabalhos desenvolvidos. Tais áreas identificadas inicialmente foram (Figura 5):

1)**Integração e Extração de Informação** – possibilitam a integração de diferentes fontes de dados, isto é, bancos de dados relacionais, arquivos de texto, HTML, XML, repositórios, etc;

2)**Ontologias para Trocas** – permuta de conhecimento que pode ocorrer com sistemas de informação legados isolados, agentes inteligentes, ou formatos de saída específicos;

3)**Adaptação e Reuso de Componentes** - encarregada do reuso e adaptação de métodos de resolução de problemas e ontologias, com o objetivo de economizar tempo e esforço de desenvolvimento;

4)**Aprendizado de Ontologias** – ao invés das ontologias terem representações estanques ou atualizações mediante usuários, também devem estar munidas de mecanismos que permitam a sua evolução, agregando

novos conceitos e relações com os conceitos já existentes;

5)**Serviços Inteligentes de Informação** - refere-se ao fornecimento de *web services* agregando algum nível de inteligência, em domínios bem específicos, tais como sistemas inteligentes de busca;

6)**Serviços de Raciocínio** – referem-se a *web services* que disponibilizariam componentes reusáveis configuráveis para sistemas de conhecimento, envolvendo basicamente métodos para resolução de problemas (PSM - Problem-Solve Methods);

7)**Web Semântica** – infra-estrutura semântica para a Web, a ser alcançada através do uso de ontologias referentes ao domínio em questão. As ontologias para a Web Semântica são baseadas em um nível fundamental no padrão XML, no qual se baseia o padrão RDF e linguagens de representação de nível mais alto (tal como OWL).

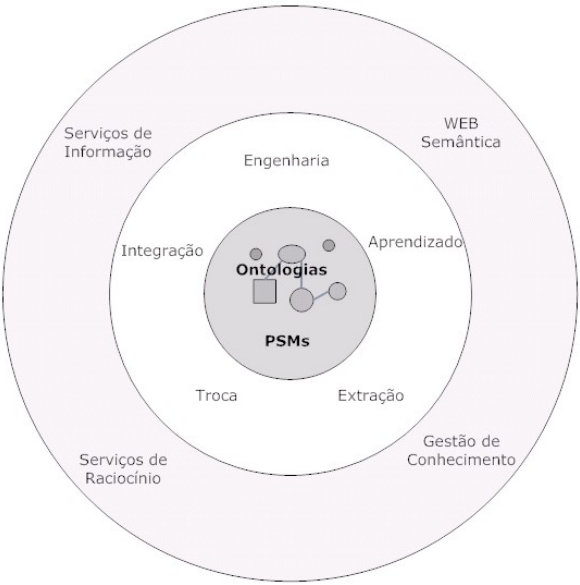


Figura 5: Métodos básicos de Engenharia do Conhecimento e áreas de aplicação.
Fonte: Adaptado de Studer et al (2000).

A Engenharia do Conhecimento também pode oferecer suporte para a manipulação de conhecimento em outras áreas, mediante o uso de ontologias, tais como os processos de aprendizagem envolvidos nas organizações (MEDEIROS et al, 2009).

2.3.2 Mecanismos de Raciocínio e Inferência

O conhecimento representado em uma ontologia mediante seus componentes (conceitos, relações, axiomas, instâncias e outros) deve servir para que novos conhecimentos sejam obtidos. Após a modelagem dos componentes de uma ontologia, deve-se lidar, portanto, com os mecanismos de inferência oferecidos pela linguagem. Assim, os chamados **motores de inferência** presentes nas plataformas de ontologias perfazem tarefas tais como i) **classificação automática de taxonomias**, ii) **gerenciamento de herança simples ou múltipla**, iii) **gerenciamento de exceções** em taxonomias de conceito; e iv) **verificação de restrições** para detectar inconsistências na ontologia ou descrever como as inferências na linguagem devem ser feitas (GÓMEZ-PÉREZ et al, 2004).

Exemplos de motores de inferência são relacionados no Quadro 5, dentre eles o LOOM classifier, presente na linguagem LOOM; o provador de teoremas JTP, implementado para Ontolingua; o motor de inferência OCML que fornece um interpretador de funções, um interpretador de controle e um sistema de prova; RDFQL para RDF; e PAL para a plataforma Protégé-2000 (GÓMEZ-PÉREZ et al, 2004).

Motores de Inferência	Descrição	Autor(es)
LOOM Classifier	Presente na linguagem LOOM, executa mecanismos de classificação e raciocínio dedutivo.	McGregor (1991) apud Gómez-Pérez (2004)
JTP- Java Theorem Prover	Provador de teoremas para ontologias escritas em Ontolingua, pode ser utilizado para dedução de informações a partir de axiomas ou para avaliação de restrições. Utiliza expressões em KIF (Knowledge Interchange Format).	Fikes et al (2003)

OCML	Motor de inferência, fornecendo um interpretador de funções, um interpretador de controle e um sistema de prova., que combina definições baseadas em frames com encadeamento para trás OCML e regras. Trabalha com checagem de restrições. Roda em interpretador Lisp.	Motta (1999)
RQL	Linguagem de consulta para RDF que adapta a funcionalidade de linguagens de consulta XML ou semiestruturadas, estendendo ainda para consultas uniformes para RDF e RDF(S)	Karvounarakis et al (2003)
PAL-Protégé Axiom Language	Checagem de restrições e valores de instâncias nas ontologias construídas em Protégé, sendo um subconjunto de KIF, podendo ser feita na aba de consultas da ferramenta	Noy et al (2000)
Lógica Anulável	Motor de inferência utilizando a chamada *lógica anulável* ou *raciocínio anulável*, uma abordagem baseada em regras para tarefas de raciocínio eficiente com informação incompleta ou inconsistente, útil na construção de ontologias onde a informação conflitante emerge naturalmente, e na modelagem de regras de negócio e políticas onde regras com exceções são muito utilizadas.	Bassiliades et al (2004)
Inference Web (IW)	Método de inferência em linguagens de marcação na Web, para suporte à Web Semântica, modelada através de uma infra-estrutura que forneça respostas mais transparentes e também um gerenciamento das explanações. Esta infra-estrutura	McGuiness e da Silva (2004)

	seria composta de uma base de conhecimento contendo regras, fontes de informação e linguagens (IW-BASE), uma linguagem de marcação de provas na forma de uma API para codificar provas portáveis; e um navegador para apresentar as provas e suas explicações (IW-BROWSE).	
DAMLJessKB	Ferramenta para raciocínio em DAML, executa inferências na Web. DAMLJessKB mapeia as marcações em DAML em fatos e regras para uso em um sistema de produção tal como o Jess (Java Expert System Shell).	Kopena e Regli (2003)
Euler	Mecanismo de inferência para OWL e RDF(S), com suporte para provas baseadas em lógica. Os axiomas são extraídos da Web e traduzidos numa espécie de programa em lógica. O motor de prova utiliza mecanismos de inferência seguindo as chamadas "trilhas de Euler".	De Roo (2003)
F-OWL	Motor de inferência para a linguagem OWL, baseado em F-logic, uma abordagem para sistemas baseados em frames em lógica.	Zou et al (2005)
Bossam	Motor de inferência para OWL, construído com base em motores de regras. Adiciona melhorias tais como ligação remota para URIs permitindo raciocínio colaborativo.	Jang e Joo-Chan (2004)
Jena, Jena2	Toolkit para RDF e OWL, utiliza uma abstração simples de grafo em RDF como sua interface interna central. Trabalha com triplas em memória, RDF em banco de dados.	Wilkinson et al (2003)

	Fornece persistência para mapeamento de triplas para banco de dados relacional.	

Quadro 5: Exemplos de motores de inferência.
Fonte: Elaborado pelo autor.

2.3.3 O Consenso em Modelagem de Ontologias

Gómez-Pérez et al (2004) declaram que o objetivo das ontologias é capturar conhecimento consensual de um domínio de maneira genérica e formal, para que seja reutilizada e compartilhada entre aplicações e grupos de pessoas. Como descrito nos conceitos de ontologias, um grupo de diversas ontologias em diferentes caminhos podem descrever um único domínio. Abordagens tais como o alinhamento e *merging* de ontologias tentam descrever um mapeamento ou unificação de ontologias no sentido de uma representação única (NOY e MUSEN, 1999). Entretanto, na maior parte das metodologias de ontologias (tais como Cyc ou KACTUS), a tarefa de conceituação ou captura de conhecimento é considerada apenas como um passo de formalização do desenvolvimento da ontologia (GÓMEZ-PÉREZ, 2004).

Em METHONTOLOGY (GÓMEZ-PÉREZ, 2004, p.131), existe a preocupação com a distância entre a percepção das pessoas e a linguagem de implementação, sendo utilizado o conceito de representações intermediárias (*intermediate representations* ou IR's). Gómez-Pérez et al (2004) adaptaram o modelo de processo essencial de Blum para o campo da engenharia ontológica. Neste modelo de processo essencial, existe uma descontinuidade entre os modelos conceituais e os modelos formalizados. Enquanto que os modelos conceituais pertencem ao nível de domínio, os modelos formalizados estão no nível de implementação. Assim, no processo de conversão de modelos conceituais para modelos formais, algum conhecimento do domínio pode ser perdido. A conclusão que se obtém a partir disto é que "**os componentes utilizados para criar modelos conceituais são mais expressivos que aqueles usados para criar modelos formais**" (GÓMEZ-PÉREZ et al, 2004, p.130).

Neste ponto, é necessário considerar o significado de "**consenso**". Em ciências de gestão, o consenso é visto como necessário para a tomada de decisão em grupo. Dado um grupo, no qual os participantes estão em competição ou em uma situação de conflito, explora-se uma variedade de alternativas (THIERAUF e KLEKAMP, 1975), mas o ato da decisão irá descartar todas as possibilidades, permanecendo apenas

aquela que foi escolhida. O consenso no grupo precisa ser alcançado, e isto significa muito menos uma unificação de diferentes opiniões do que um processo de descarte, onde apenas uma opinião irá prevalecer. Portanto, este conceito de consenso possui uma forte analogia com o problema do modelo conceitual-formal exposto no parágrafo anterior, se considerado o problema do desenvolvimento da ontologia como um problema de decisão sobre quais classes ou relações (de um modelo conceitual) serão "congelados" em uma ontologia formalizada. A solução para este problema de "poder de representatividade" precisa ser elaborada a partir de tipos específicos de representação de ontologias que possam ter um desempenho melhor no limiar dos modelos conceitual-formal, antes de ter lugar o processo de consenso.

Esta fundamentação sobre ontologias buscou mostrar seus conceitos principais associados ao seu estudo, bem como as atividades, ferramentas, aspectos de construção, e mecanismos de inferência, e também o relacionamento e a importância para a área da Engenharia do Conhecimento. Alguns conceitos são retomados em profundidade na descrição do *framework* considerando a Computação Quântica e as subáreas derivadas. Uma discussão relativa ao consenso e modelos conceitual-formal é retomada quando apresentado o tópico da superposição de classes.

2.4 COMPUTAÇÃO QUÂNTICA

2.4.1 Breve Histórico

A Computação e Informação Quântica nasceram a partir da ideia da simulação de experimentos físicos levados a efeito em computadores. É possível que computadores normais consigam simular a realidade dos experimentos físicos, a partir de modelos construídos que espelhem fielmente a realidade? Paul Benioff (1980) expressou as primeiras ideias relacionando a Mecânica Quântica à Computação com referência a máquinas de Turing que pudessem ser simuladas explorando modelos de hamiltonianos[6] da Mecânica Quântica. Richard Feynman (1982) então questionava se os computadores universais poderiam efetivamente simular os eventos físicos. Usualmente os cientistas imitam o mundo clássico descrevendo o mesmo através de soluções aproximadas de

[6] Na Mecânica Quântica, constituem matrizes de estados de energia que explicam a evolução de um sistema quântico, de acordo com a equação de Schröedinger.

equações diferenciais. Na visão de Feynman, o mundo real é o da Mecânica Quântica. Se tal mundo pudesse ser simulado, ele teria de ser **exatamente** simulado. Para sistemas de uma partícula apenas, a simulação de probabilidades dada por uma equação de onda de Schröedinger pode ser executada de forma trivial. O problema da simulação se agrava quando se adiciona muitas partículas ao sistema quântico. A quantidade de variáveis a serem controladas acaba sendo demasiadamente grande, e um computador normal não pode simular a realidade para tal conjunto de elementos quânticos. Para que isso acontecesse, o computador no qual seria feita a simulação deveria ser construído de acordo com as leis da Mecânica Quântica.

Feynman vai além, e afirmou que, mesmo uma simulação não sendo exata, mas probabilística, o computador clássico ainda não teria condições de fazer tal simulação quântica. Pode-se evocar o problema de existirem *variáveis ocultas*, que não podem ser representadas fisicamente e não podem, por sua vez, ser representadas num computador clássico. E também quanto à questão da existência dos efeitos não-locais, que não podem ser simulados por um computador clássico local (FEYNMAN, 1982). Entretanto, com relação à presença de efeitos não-locais entre objetos quânticos, os trabalhos mais significativos publicados foram o de Freedman e Clauser (1972) e o de Aspect, Grangier e Roger (1982). Estes pesquisadores realizaram em laboratório uma versão do experimento EPR, com resultados que violavam as desigualdades de Bell e confirmando os resultados previstos pela Mecânica Quântica, quanto à não-localidade dos eventos quânticos. Algumas propriedades de sistemas quânticos tais como os distúrbios inevitáveis envolvidos na medida foram visualizados como tendo uso prático em criptografia quântica (STEANE, 1997; BENNETT e BRASSARD et al, 1984). Porém, o salto significativo em direção à Computação Quântica foi dado por David Deutsch, sendo o primeiro a explorar as possibilidades de fazer computação com sistemas de mecânica quântica, propondo um análogo à máquina de Turing denominado de **Computador Quântico Universal**. Neste trabalho ele ainda propôs a primeira versão do que hoje é denominado de algoritmo de Deutsch, abrindo caminho para o desenvolvimento de algoritmos mais complexos e um desenvolvimento maior da Computação Quântica. As operações simples utilizadas por ele no algoritmo são denominadas agora de **portas quânticas**. Deutsch se baseou no princípio da superposição quântica em seu algoritmo, explorando o aspecto do paralelismo massivo (DEUTSCH, 1985).

Outra característica de um sistema quântico concebido como um circuito é a possibilidade de **reversibilidade** das operações. Ainda que a maior parte do desenvolvimento da Computação Quântica tenha acontecido na década de 1980, Charles Bennett já havia explorado anteriormente esta possibilidade da máquina de Turing fazendo computação utilizando processos reversíveis (BENNETT, 1973). As portas lógicas utilizadas nos circuitos eletrônicos comuns não poderiam fornecer reversibilidade, e certas portas lógicas como a porta Toffoli e a porta Fredkin foram então concebidas de forma a permitir a computação reversível (TOFFOLI, 1980; FREDKIN e TOFFOLI, 1982).

O algoritmo de Deutsch, como foi proposto inicialmente, utilizava apenas dois q-bits. Em trabalho subseqüente, este algoritmo foi aprimorado de forma a conter mais q-bits, mostrando que sistemas de Computação Quântica poderiam ser **escaláveis** (DEUTSCH e JOSZA, 1992), sendo conhecido então como o algoritmo de Deutsch-Josza. Foi demonstrado então que alguns problemas computacionais poderiam ser resolvidos de forma mais eficiente em um computador quântico do que em um computador clássico (BERTHIAUME e BRASSARD, 1992; BERNSTEIN e VAZIRANI, 1993). Um importante avanço foi feito por Simon (1994) que descreveu um algoritmo quântico eficiente para o qual não há uma solução eficiente pela abordagem clássica, mesmo utilizando métodos probabilísticos, explorando a periodicidade em aritmética modular. Este trabalho inspirou Peter Shor (1994) a descrever um algoritmo o qual não era apenas eficiente em um computador quântico, mas envolvia um problema fundamental em ciência da computação, a **fatoração** de números primos muito grandes. A força do sistema de criptografia RSA, muito utilizado por várias corporações mundiais, reside no fato de que não é possível fatorar em tempo polinomial o número composto fornecido para a chave pública de 128 ou 256 bits. Entretanto, Shor demonstrou que com um computador quântico, contendo um número suficiente de q-bits, seria possível a descoberta da chave privada do sistema RSA através deste algoritmo, o qual mesmo sendo probabilístico é **eficiente** (SHOR, 1994). O algoritmo de Simon e o algoritmo de Shor fazem uso das propriedades de superposição e emaranhamento quântico.

Junto com o algoritmo de Shor, outro algoritmo que mostrou a força da Computação Quântica em relação à Computação Clássica foi o **algoritmo quântico de busca** desenvolvido por Lov Grover (1996). Mediante o uso da superposição e interferência quântica, Grover mostrou que a busca de um item num conjunto de elementos não

ordenados poderia ser feita com complexidade menor do que num algoritmo clássico. Mais tarde foi demonstrado que o algoritmo de Grover perfaz uma busca ótima (ZALKA, 1999).

Devido ao fato da Computação Quântica requerer computadores quânticos para a efetiva execução, tais computadores estão sujeitos a efeitos de ruído do ambiente que podem interferir na precisão dos resultados dos algoritmos. A computação quântica deve ser feita com um grau razoável de acurácia, e conseguir isto é uma tarefa muito mais desafiadora em relação a computadores clássicos (NAGY e AKL, 2005). O estudo da **correção quântica de erros**, unindo os conceitos da Computação Quântica com a Teoria da Informação Clássica foi então considerado para contornar o problema da presença de ruído ou efeitos indesejáveis nos circuitos quânticos (PRESKILL, 1998). Calderbank e Shor (1996) e Steane (1996) estabeleceram um esquema genérico onde o processamento da informação quântica pudesse ser utilizado para combater uma grande classe de processos de ruído em um sistema quântico propriamente desenhado para isto (STEANE, 1997). Este esquema incluía circuitos com q-bits adicionais para garantir a fidelidade do processamento quântico no circuito principal.

Numa linha diferente dos algoritmos de Shor e de Grover, que iniciam com uma superposição de estados, outra proposta de algoritmo para efetuar busca combinatorial considerando satisfação de restrições foi feita por Tad Hogg (1996). As soluções seriam obtidas a partir de um mapeamento de variáveis, onde existiria um reticulado natural dado pelo conjunto da combinação das variáveis. Hogg também apresentou dois métodos para mover-se sobre o reticulado, aumentando-se a amplitude dos conjuntos para superconjuntos e assim sucessivamente, com isto sendo feito a partir de matrizes unitárias. A ideia central do algoritmo de busca de Hogg é fazer com que a busca aconteça distanciando-se cada vez mais dos conjuntos que violam as restrições (HOGG, 1996; RIEFFEL e POLAK, 2000).

O **teorema da não-clonagem** foi descoberto por Dieks (1982) e Wooters e Zurek (1982). Este teorema levanta um dos aspectos fundamentais dentro da Computação Quântica, onde não se pode copiar um estado de um q-bit para outro, a não ser que sejam os estados da base. Ainda que a clonagem de estados quânticos arbitrários não seja possível, o **teleporte quântico** foi proposto por Bennett et al (1993) como forma de se transportar um estado quântico desconhecido utilizando a propriedade do emaranhamento. O teleporte quântico é uma técnica utilizada para tal transporte de estados ainda que não exista um

canal de comunicação conectando o transmissor do estado ao seu receptor. O teleporte quântico permite a possibilidade da *codificação superdensa*, sendo que um par emaranhado EPR mais dois bits de informação clássica equivalem a um q-bit de comunicação quântica (NIELSEN e CHUANG, 2005).

Boa parte dos algoritmos, tais como o algoritmo de Deutsch, Deutsch-Josza e o de Grover, serviram como ponto de partida para as primeiras implementações práticas. Uma característica relevante da construção de tais implementações é a não fixação sobre um único **paradigma de hardware**, sendo possível a realização física de várias maneiras: íons aprisionados, fótons, ressonância magnética nuclear (RMN) ou ainda usando cavidades de eletrodinâmica quântica. Tais implementações são descritas em maior profundidade no Apêndice A, relativo ao hardware quântico.

Uma variante da Computação Quântica foi baseada no teorema de evolução adiabática, por isso sendo denominada de **Computação Quântica Adiabática**. Neste modelo, a evolução de um estado quântico é governada por uma matriz de estados de energia dependente do tempo, que interpola de um estado inicial ao estado final (FARHI et al, 2000). Ao invés de se construir um circuito quântico contendo várias portas para mudar o estado inicial em direção ao estado final passo-a-passo, a evolução se daria de forma contínua e suave até o alcance do estado final. Entretanto, Aharonov et al (2004) mostraram que o algoritmo tratado por Farhi et al (2000) levaria tempo exponencial para a resolução do problema tratado (3-SAT) para o pior caso, e que o modelo de Computação Quântica Adiabática seria equivalente ao modelo padrão de Computação Quântica.

Apesar de certos algoritmos terem sido desenvolvidos aproveitando-se o potencial de propriedades quânticas tais como a superposição e o emaranhamento, alguns estudos tem mostrado a **impossibilidade** física de se executar certos algoritmos clássicos em computadores quânticos. Por exemplo, Lomont (2000) demonstrou que algoritmos de correlação ou convolução, técnicas largamente utilizadas na construção de filtros digitais, são impossíveis de se realizar fisicamente em estados quânticos.

Na Figura 6 são mostrados alguns dos principais eventos relacionados ao desenvolvimento da área da Computação e Informação Quântica mostrando, além de estudos teóricos, algumas das implementações experimentais feitas com computadores quânticos em diversos paradigmas.

2.4.2 O Q-Bit

A abordagem a ser seguida na explicação dos elementos de Computação Quântica, nesta parte da fundamentação teórica, é baseada predominantemente em Nielsen e Chuang (2005), Perry (2004), Preskill (1998) e Steane (1997). A Computação Quântica difere da Computação Clássica quanto ao próprio bloco básico de informação, o qual é visto de maneira diferente do elemento clássico, o *bit*. Este segue a lógica clássica, somente podendo assumir dois valores, 0 e 1. Na Computação Quântica, o elemento básico a ser tratado é o **q-bit** ou **qubit** (*quantum bit*, bit quântico). Enquanto o bit clássico pode somente assumir dois estados, 0 e 1, o q-bit pode assumir infinitos estados na forma de superposições de 0's e 1's (NIELSEN e CHUANG, 2005). Para a representação de q-bits, utiliza-se a **notação de Dirac** (|.⟩), tal como na Mecânica Quântica[7]. Um q-bit recebe a mesma representação de uma função de onda na Mecânica Quântica, e os estados que este q-bit pode assumir são |0⟩ ou |1⟩ (de forma análoga à Computação Clássica), ou ainda uma combinação linear entre estes estados. O espaço de estados quânticos possíveis, no qual os estados |0⟩ e |1⟩ são apenas duas opções, é o **espaço de Hilbert**. Uma forma de entender o significado de um estado quântico na notação de Dirac é utilizar uma representação vetorial em forma de vetor coluna para os estados,

$$|0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

e

$$|1\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Os estados |0⟩ e |1⟩ definem uma **base vetorial** para o espaço de Hilbert de um q-bit. Portanto, um estado quântico qualquer |ψ⟩ pode ser representado mediante uma combinação linear destes estados da base,

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$$

com α e $\beta \in X^2$. Representando esta expressão na forma vetorial,

[7] Esta notação é denominada de *ket* por ser a metade de um *bracket* (<.|.>).

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} \alpha \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ \beta \end{bmatrix} = \alpha \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \beta \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Entretanto, não temos acesso ao conteúdo de um q-bit, ou seja, não podemos saber quais os valores exatos[8] de α e β. De acordo com o **postulado da medição** da Mecânica Quântica, a observação do estado quântico causa o colapso da função de onda. Quando o q-bit é medido, pode-se obter apenas o estado $|0\rangle$ com probabilidade $|\alpha|^2$, e o estado $|1\rangle$ com probabilidade $|\beta|^2$. As probabilidades somadas devem resultar em 1, ou seja,

$$|\alpha|^2 + |\beta|^2 = 1$$

Por exemplo, o estado quântico **normalizado** correspondente às probabilidades iguais entre os estados $|0\rangle$ e $|1\rangle$, ou seja, 50% para cada estado, é definido pela seguinte combinação linear,

$$|\psi\rangle = \frac{|0\rangle + |1\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}} \begin{bmatrix} 1 \\ 1 \end{bmatrix},$$

com $\alpha = \beta = 1/\sqrt{2}$. Este exemplo ajuda a ilustrar o **princípio da superposição** de estados quânticos. Até que se meça o estado $|\psi\rangle$, ou seja, enquanto não se observa tal estado, ele é ao mesmo tempo os dois estados, $|0\rangle$ e $|1\rangle$, combinados linearmente.

[8] α e β são denominados também *coeficientes* dos estados, e podem assumir valores reais ou complexos (números imaginários). Esta caracterização de números complexos nos coeficientes torna difícil a visualização da função de onda ou estado como algo possuindo realidade física. A representação em forma de probabilidades permite uma visualização, devido ao resultado do quadrado do módulo do coeficiente de um número complexo produzir sempre um número real.

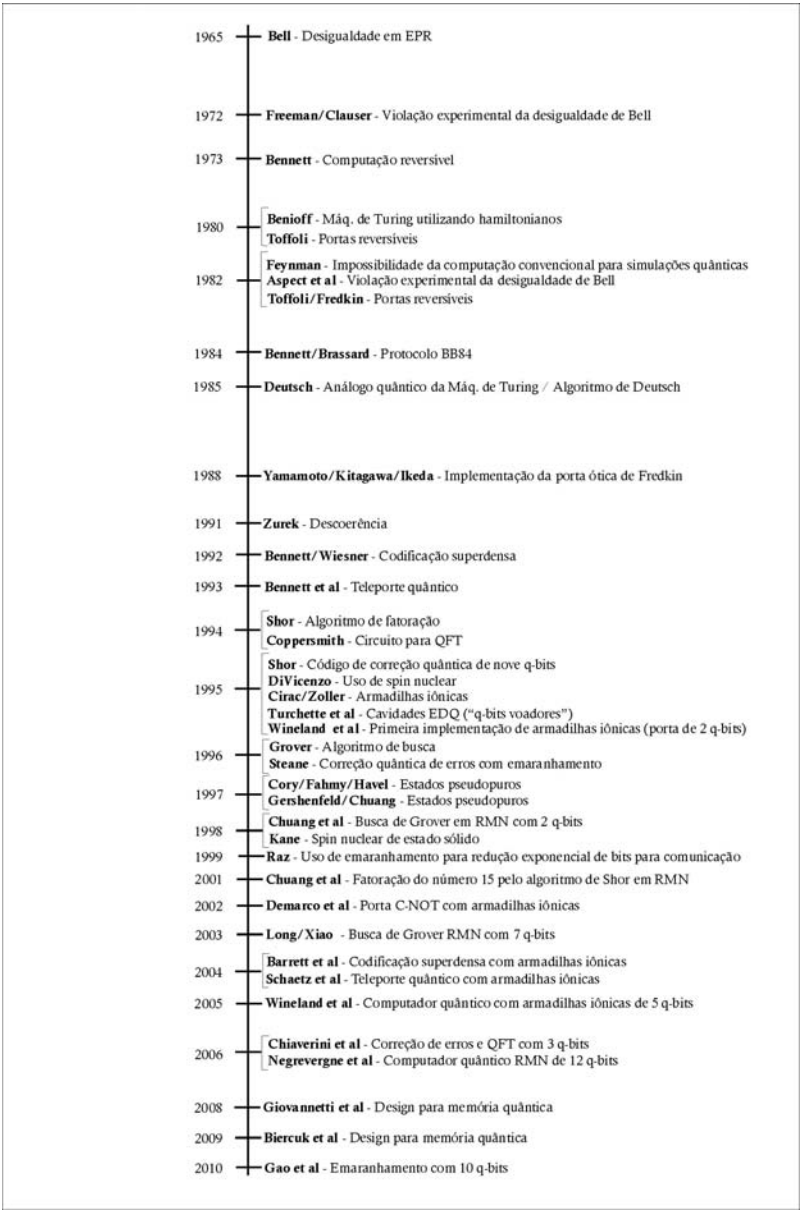


Figura 6: Linha do tempo com principais eventos relacionados ao desenvolvimento da Computação e Informação Quântica. Fonte: Elaborado pelo autor.

Nielsen e Chuang (2005) ressaltam que, normalmente nos modelos abstratos sobre o mundo, existe uma correspondência entre os elementos do mundo real e os da abstração. No mundo quântico, não é possível esta correspondência direta, e isto torna difícil a intuição do comportamento de objetos quânticos. Porém, é possível uma **correspondência indireta**, pois os estados dos q-bits podem ser manipulados para direcionar os resultados de medidas, que estarão na dependência de diferentes estados. Portanto, tais estados quânticos terão conseqüências reais que podem ser **verificáveis experimentalmente**. O q-bit possui assim dois comportamentos: um quando não se observa, correspondendo ao mundo quântico; e outro após a detecção ou medida do estado que colapsa a função de onda.

Qual a quantidade de informação que pode armazenar um q-bit? A princípio, parece que existe um número infinito de informação que poderia ser armazenada, devido à propriedade de superposição. Mas após a medida, **apenas um estado** é assumido pelo q-bit. Apesar disso, uma questão ainda pode ser levantada: quanta informação pode ser armazenada em um q-bit se não for medido? Porém, de que maneira pode-se quantificar uma informação se não é possível medi-la? O importante a considerar neste aspecto é que, ainda que não se realize medidas, **a natureza faz evoluir um sistema de q-bits**, aparentemente acompanhando todas as variáveis que descrevem o sistema, assim como α e β. Ao que parece, uma grande quantidade de informação oculta sobre o estado de um q-bit é manipulado pela natureza, e parece que aumenta exponencialmente se aumentarmos o número de q-bits envolvidos (NIELSEN e CHUANG, 2005).

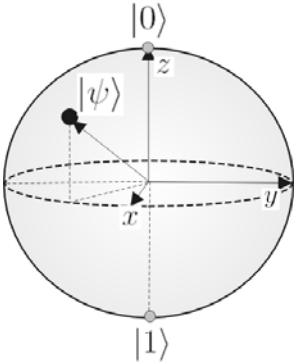


Figura 7: Esfera de Bloch com a representação de um q-bit.
Fonte: Adaptado de Nielsen e Chuang (2005).

Uma forma geométrica de se visualizar um q-bit é mostrado na Figura 7, onde é mostrada a **esfera de Bloch**. Sobre a superfície desta esfera, um q-bit pode assumir infinitos valores (PERRY, 2006; NIELSEN e CHUANG, 2005). Porém, depois da respectiva medição, os únicos valores finais são os pontos representados pelos estados $|0\rangle$ e $|1\rangle$.

2.4.3 Combinação de q-bits

Assim como os bits clássicos são combinados em registradores para aumentar a faixa de valores binários possíveis, q-bits podem ser combinados também em **registradores** nos circuitos quânticos. Para um sistema clássico com dois bits, os valores possíveis resultariam da combinação entre eles: 00, 01, 10 e 11. De forma equivalente, um sistema quântico de dois q-bits irá possuir 4 estados**:** $|00\rangle$, $|01\rangle$, $|10\rangle$ e $|11\rangle$. Assim como um q-bit pode estar em um estado de superposição de dois estados, o sistema duplo de q-bits estaria em uma superposição dos quatro estados respectivos:

$$|\psi\rangle = \gamma_{00}|00\rangle + \gamma_{01}|01\rangle + \gamma_{10}|10\rangle + \gamma_{11}|11\rangle$$

Um sistema quântico com dois ou mais q-bits utiliza a operação do **produto tensorial** para combinar os estados quânticos. O produto tensorial entre estados quânticos é representado por pelo sinal $\otimes$ (AHARONOV, 1998; EKERT et al, 2000). Utilizando novamente a representação vetorial para os estados quânticos envolvendo dois q-bits com as respectivas operações de produto tensorial:

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$|01\rangle = \begin{bmatrix}1\\0\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$$

$$|10\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}1\\0\end{bmatrix} = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$$

$$|11\rangle = \begin{bmatrix}0\\1\end{bmatrix} \otimes \begin{bmatrix}0\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$$

Quando uma medida se realiza sobre este sistema de dois q-bits, o resultado será um estado apenas dentre as quatro alternativas possíveis $|ij\rangle$, com $i,j=\{0,1\}$, e a probabilidade respectiva $|\gamma_{ij}|^2$.

Pode-se escolher medir o estado de apenas um q-bit do conjunto, digamos, o primeiro q-bit. Pode-se verificar que a medida deste primeiro q-bit irá fornecer o estado $|0\rangle$, com uma probabilidade $|\gamma_{00}|^2 + |\gamma_{01}|^2$. Após a medição, o segundo q-bit pode assumir qualquer um dos estados, $|0\rangle$ ou $|1\rangle$. Portanto, o estado final $|\psi'\rangle$ do q-bit será

$$|\psi'\rangle = \frac{\gamma_{00}|00\rangle + \gamma_{01}|01\rangle}{\sqrt{|\gamma_{00}|^2 + |\gamma_{01}|^2}}$$

O estado quântico após a medida deve ser normalizado, sendo dividido por $\sqrt{|\gamma_{00}|^2 + |\gamma_{01}|^2}$.

Certas representações de dois ou mais q-bits podem existir sem que tenham sido produzidos através de um produto tensorial. Por exemplo, pode-se supor que dois bits estejam correlacionados quanticamente, num estado **emaranhado**. O **emaranhamento** é uma

das propriedades mais estranhas no mundo quântico. Um estado tal como o que segue para os dois q-bits é possível:

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Este estado faz parte de um conjunto denominado **estados de Bell**[9] ou **estados EPR**[10]. O estado anterior é relacionado com o experimento EPR, onde duas partículas estão correlacionadas, sendo que a medição em uma partícula irá causar o colapso simultâneo na outra. Deve-se notar que, nos dois estados apresentados, os q-bits encontram-se iguais: quando o primeiro q-bit é $|0\rangle$, o segundo também é; quando o primeiro q-bit é levado ao estado $|1\rangle$, o segundo q-bit também vai ao estado $|1\rangle$. Este estado de Bell ou estado EPR tem a propriedade de originar, após a medida, um estado $|00\rangle$ com probabilidade de 1/2; e o estado $|11\rangle$ também com probabilidade[11] de 1/2. Tais estados **não podem ser resultantes da operação do produto tensorial** entre q-bits individuais. A representação vetorial da expressão (8) é dada pela expressão a seguir,

$$\frac{|00\rangle + |11\rangle}{\sqrt{2}} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\0\\0\\1\end{bmatrix}$$

John Bell (1964) mostrou que as correlações observadas nas medições de pares EPR "são maiores que quaisquer outras que poderiam existir em sistemas clássicos" (NIELSEN e CHUANG, 2005). Portanto, tais resultados mostrariam que a computação quântica poderia fornecer resultados que iriam **além do escopo** da computação clássica.

---

[9] Devido a John Bell (1928-1990), que estudou em profundidade o experimento EPR, propondo equações de desigualdades que deveriam provar ou refutar a mecânica quântica.

[10] Iniciais de Einstein, Podolsky e Rosen, que argumentaram contra a completitude da Mecânica Quântica apresentando um experimento de pensamento com duas partículas correlacionadas.

[11] Ou seja, os dois estados têm igual probabilidade de 50%.

Para demonstrar o poder da computação quântica, Nielsen e Chuang (2005) fazem ainda uma expansão interessante: para sistemas com *n* q-bits, os estados possíveis serão descritos por $|x_1x_2x_3...x_n\rangle$. Portanto, o estado quântico de um sistema com *n* q-bits irá conter 2^n estados. Para o caso onde *n*=500, este número é bem maior do que o número de átomos existentes no Universo. É impossível, portanto, conceber fisicamente um computador clássico que tenha condições de armazenar toda esta quantidade de informação. Porém, a natureza é capaz de manipular esta grande quantidade de variáveis e fazer evoluir o estado de tal sistema quântico[12].

No Quadro 6 são enumeradas algumas das principais diferenças entre a Computação Clássica e Quântica.

Item	Computação Clássica	Computação Quântica
Elemento básico de representação	Bit.	Q-bit.
Domínio de valores	Apenas dois estados lógicos (0 e 1).	Domínio contínuo de estados com coeficientes complexos.
Superposição	Não existe, os valores assumidos são bem definidos.	O q-bit, enquanto não observado, pode assumir uma superposição de estados quânticos.
Determinismo	Os estados lógicos são assumidos com certeza total.	Antes da medição, o estado quântico evolui de forma determinística. O estado de um q-bit é probabilístico, após o ato da medição.
Dependência	Os estados lógicos de um bit são	Os estados quânticos de um q-bit podem se emaranhar,

[12] Nielsen e Chuang argumentam que "é como se a Natureza mantivesse 2^{500} folhas de papel de rascunho escondidas, nas quais ela realiza seus cálculos à medida que o sistema evolui".

	independentes.	de forma que a mudança de estado em um q-bit pode alterar o estado de outro q-bit.
Medida	A medida de um determinado estado lógico não altera o estado de outros bits do sistema	A medida do estado quântico de um q-bit altera o estado final, isolado ou correlacionado com outros q-bits.

Quadro 6: Comparação entre Computação Clássica e Quântica.
Fonte: Elaborado pelo autor.

2.4.4 Produto Interno e Ortonormalidade

A notação de Dirac permite ainda a definição do vetor **dual**. Assim como um estado quântico pode ser representado na forma de **vetor coluna** ou **ket**

$$|\psi\rangle = \begin{bmatrix} \alpha \\ \beta \end{bmatrix},$$

existe o **vetor linha**, que na notação de Dirac pode ser explicado pela forma matricial $\langle\psi| = [\alpha^* \quad \beta^*]$. O vetor $\langle\psi|$, também denominado de **bra**, não é simplesmente o vetor coluna transposto para linha, mas as amplitudes α^* e β^* são **conjugados complexos**. A noção de vetor dual permite a definição do **produto interno** entre dois vetores ou estados $|\psi\rangle$ e $|\phi\rangle$ como sendo $\langle\psi|\phi\rangle$, onde $\langle\psi| = (|\psi\rangle)^{\dagger}$. Por exemplo, tendo-se o vetor coluna com os seguintes componentes,

$$|\phi\rangle = \begin{bmatrix} \gamma \\ \delta \end{bmatrix}$$

O produto interno entre os vetores $|\psi\rangle$ e $|\phi\rangle$ é dado por

$$\langle\psi|\phi\rangle = \begin{bmatrix} \alpha^* & \beta^* \end{bmatrix} \begin{bmatrix} \gamma \\ \delta \end{bmatrix} = \alpha^*\gamma + \beta^*\delta .$$

O produto interno tem como resultado um número escalar, com $\langle\psi|\phi\rangle \in X$, ao contrário do produto tensorial, que resulta em outro vetor de dimensões mais altas no espaço de Hilbert.

A definição de **ortogonalidade** entre vetores pode ser obtida a partir do produto interno. Se $|\psi\rangle$ e $|\phi\rangle$ são dois vetores distintos no espaço de Hilbert, tais vetores são ortogonais se e somente se

$$\langle \psi \mid \phi \rangle = 0 .$$

Os vetores ortogonais permitem a definição de uma **base** para a representação no espaço de Hilbert. Como visto no início da seção, os estados $|0\rangle$ e $|1\rangle$ constituem uma base para representação vetorial no espaço de Hilbert, pois fazendo-se o produto interno entre os dois estados,

$$\langle 0 \mid 1 \rangle = \langle 1 \mid 0 \rangle = 0$$

Porém, a condição de ortogonalidade é necessária, mas não é suficiente para a formação de uma base. Os estados precisam ainda ser **normalizados**. O produto interno permite o cálculo da **norma** de um vetor:

$$|| \psi || = \sqrt{\langle \psi \mid \psi \rangle} .$$

Um vetor está normalizado se ele é um vetor **unitário**, ou seja, se a sua norma $||\psi||=1$. A normalização de um vetor é feita dividindo-se o vetor por sua norma:

$$\frac{|\psi\rangle}{||\psi||} .$$

Pode-se verificar que a base ($|0\rangle$, $|1\rangle$) possui vetores unitários, ou seja,

$$\langle 0 \mid 0 \rangle = \langle 1 \mid 1 \rangle = 1 .$$

Portanto, a base de estados ($|0\rangle$, $|1\rangle$) que possui as duas condições, a de ortogonalidade e normalização, é dita **ortonormal**. Em Mecânica Quântica e Computação Quântica, apenas com a adoção de uma base ortonormal é que se pode distinguir (medir) os estados quânticos de um sistema (NIELSEN e CHUANG, 2005).

2.4.5 Portas Quânticas

A manipulação de estados dos q-bits em um circuito quântico deve ser feito através de **portas quânticas**. Na Computação Clássica, existem portas digitais para manipular bits em um circuito, tal como uma porta NOT que inverte um bit na sua entrada, ou a porta AND, a qual só produz o bit 1 na saída se e somente se os dois bits de entrada forem 1 (Figura 8). Da mesma forma, uma porta quântica deve atuar sobre o estado de um q-bit, transformando a sua saída para outro estado (NIELSEN e CHUANG, 2005).

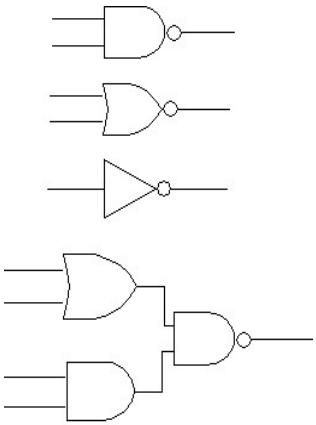


Figura 8: Exemplos de portas digitais.
Fonte: Elaborado pelo autor.

As portas quânticas básicas são aquelas que atuam sobre um único q-bit. Na Mecânica Quântica existem os **operadores de Pauli**, que fornecem a base para as portas de um q-bit utilizadas na Computação Quântica, denotadas pelos símbolos ***X***, ***Z*** e ***Y***. No Quadro 7 está ilustrada a operação que cada porta faz com um vetor estado de entrada $|\psi\rangle$.

Símbolo	**Representação**	**Operador**	**Resultado sobre $\|\psi\rangle=\alpha\|0\rangle+\beta\|1\rangle$**
X ou $\boldsymbol{\sigma_x}$		$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$	$\|\psi'\rangle=\beta\|0\rangle+\alpha\|1\rangle$

Z ou σ_z		$\begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$	$\|\psi'\rangle=\alpha\|0\rangle-\beta\|1\rangle$
Y ou σ_y		$\begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}$	$\|\psi'\rangle=-i\,\beta\|0\rangle+i\,\alpha\|1\rangle$

Quadro 7: Portas quânticas dos Operadores de Pauli
Fonte: Elaborado pelo autor.

A porta ***X*** funciona como uma porta inversora, trocando os coeficientes dos estados quânticos. A porta ***Z*** faz com que o estado $|1\rangle$ seja transformado em $-|1\rangle$. A porta ***Y*** muda os coeficientes para o eixo complexo. De acordo com o formalismo do espaço vetorial de Hilbert, uma porta quântica funciona na verdade como um **operador linear**, e o novo estado do q-bit é obtido multiplicando-se este operador pelo estado atual do q-bit. A atuação da porta ***X*** sobre o estado $|\psi\rangle$ pode ser vista como a seguinte operação matricial:

$$X\,|\psi\rangle=\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}\begin{bmatrix} \alpha \\ \beta \end{bmatrix}=\begin{bmatrix} \beta \\ \alpha \end{bmatrix}$$

Símbolo	Representação	Operador	Resultado sobre $\|\psi\rangle=\alpha\|0\rangle+\beta\|1\rangle$
H		$\frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$	$\|\psi'\rangle=\alpha\frac{\|0\rangle+\|1\rangle}{\sqrt{2}}+\beta\frac{\|0\rangle-\|1\rangle}{\sqrt{2}}$
S		$\begin{bmatrix} 1 & 0 \\ 0 & i \end{bmatrix}$	$\|\psi'\rangle=\alpha\|0\rangle+i\,\beta\|1\rangle$
T		$\begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}$	$\|\psi'\rangle=\alpha\|0\rangle+e^{i\pi/4}\,\beta\|1\rangle$

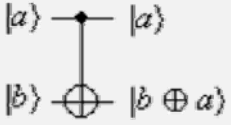
Quadro 8: Operadores das portas *H*, *S* e *T*
Fonte: Elaborado pelo autor.

No Quadro 8 estão representadas as portas ***H***, ***S*** e ***T***. A porta ***H*** é denominada **porta Walsh-Hadamard**, e sua aplicação em um estado puro, $|0\rangle$ ou $|1\rangle$, produz uma superposição de estados para o vetor resultante. Por exemplo, aplicando-se a porta ***H*** sobre o vetor $|0\rangle$,

$$H\,|\,0\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix}1 & 1\\ 1 & -1\end{bmatrix}\begin{bmatrix}1\\ 0\end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix}1\\ 1\end{bmatrix} = \frac{|\,0\rangle + |\,1\rangle}{\sqrt{2}}$$

A porta ***S*** também é denominada **porta de fase**. A porta ***T*** é denominada também **porta π/8**.

Portas quânticas também podem atuar sobre dois q-bits ou mais. A porta quântica de dois q-bits mais comum é a **porta C-NOT**, ou NOT-controlado (NIELSEN e CHUANG, 2005; EKERT et al, 2000). A porta C-NOT utiliza um q-bit de alvo mais um q-bit de controle. A regra é simples: quando o q-bit de controle assume o valor $|1\rangle$, o estado do q-bit alvo é invertido. No Quadro 9 é mostrada a operação C-NOT sobre q-bits. Note que, para dois q-bits, o operador linear é uma matriz 4x4. A representação do vetor no estado final mostra os dois q-bits combinados na forma $|a\rangle|b\rangle$, com $|a\rangle$ sendo o q-bit de controle, e $|b\rangle$ o q-bit alvo. Outra forma de representar a saída da operação C-NOT é utilizar a operação de OU-exclusivo ou adição de módulo 2 ($\oplus$) entre os q-bits de controle e alvo ($0 \oplus 0 = 0$; $0 \oplus 1 = 1$; $1 \oplus 0 = 1$; $1 \oplus 1 = 0$).

Símbolo	**Representação**	**Operador**	$\|\psi\rangle=\|a\rangle\|b\rangle \rightarrow \|a\rangle\|b \oplus a\rangle$ ***a*** **= q-bit de controle** ***b*** **= q-bit alvo**
C-NOT		$\begin{bmatrix}1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 1 & 0\end{bmatrix}$	$\|00\rangle \rightarrow \|00\rangle$ $\|01\rangle \rightarrow \|01\rangle$ $\|10\rangle \rightarrow \|11\rangle$ $\|11\rangle \rightarrow \|10\rangle$

Quadro 9: Operador CNOT.
Fonte: Elaborado pelo autor.

A porta C-NOT, em conjunto com as portas de um q-bit, constitui um **conjunto universal** de portas quânticas, devido ao fato de que qualquer circuito quântico pode ser simulado utilizando-se este conjunto universal (BARENCO et al, 1995).

As portas quânticas, visualizadas pelo aspecto da representação matricial, estão sujeitas a algumas propriedades peculiares. É o caso de algumas possuírem sua **matriz adjunta**. A matriz adjunta é interpretada como a matriz conjugada transposta. Como os elementos da matriz pertencem a X, o conjunto dos números complexos, eles admitem valores conjugados:

$$(a+bi)^* = (a-bi)$$

Assim, uma matriz $A^{\surd}$ é dita adjunta caso tenha seus elementos dispostos da seguinte forma

$$\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix}^{\mathrm{T}}\right)^* = \begin{bmatrix} a^* & c^* \\ b^* & d^* \end{bmatrix},$$

ou seja,

$$A^{+} = \left(A^{\mathrm{T}}\right)^*.$$

As portas ***X***, ***Z*** e ***H*** são iguais às suas adjuntas, sendo também denominadas de **hermitianas**. As portas ***Y***, ***S*** e ***T*** possuem adjuntas diferentes, estando as mesmas representadas no Quadro 10.

Símbolo	Representação	Operador	Resultado sobre $\|\psi\rangle=\alpha\|0\rangle+\beta\|1\rangle$
$Y^{\surd}$		$\begin{bmatrix} 0 & i \\ -i & 0 \end{bmatrix}$	$\|\psi'\rangle=i\,\beta\|0\rangle-i\,\alpha\|1\rangle$
$S^{\surd}$		$\begin{bmatrix} 1 & 0 \\ 0 & -i \end{bmatrix}$	$\|\psi'\rangle=\alpha\|0\rangle-i\,\beta\|1\rangle$
$T^{\surd}$		$\begin{bmatrix} 1 & 0 \\ 0 & e^{-i\pi/4} \end{bmatrix}$	$\|\psi'\rangle=\alpha\|0\rangle+e^{-i\pi/4}\,\beta\|1\rangle$

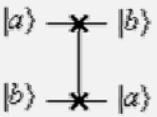
Quadro 10: Operadores adjuntos das portas *Y*, *S* e *T*.
Fonte: Elaborado pelo autor.

Quando um dos operadores de Pauli ou a porta Walsh-Hadamard são aplicados novamente em um circuito onde já exista um operador igual, produzem como saída a matriz identidade, ou seja,

$$XX\,|\psi\rangle = I\,|\psi\rangle = |\psi\rangle\ .$$

A porta C-NOT utilizando dois q-bits também apresenta esta propriedade. A aplicação dupla equivale à **inversão de direção** na evolução do circuito, demostrando o aspecto da **reversibilidade** dos circuitos quânticos.

Outra aplicação da porta C-NOT é o circuito de troca ou **porta de troca** (SWAP), cuja representação é mostrada no Quadro 11. Combinando-se três portas C-NOT de acordo com a Figura 9 têm-se a porta de troca (SWAP), que faz a troca dos estados entre dois q-bits.

Símbolo	**Representação**	**Operador**	$\|\psi\rangle=\|a\rangle\|b\rangle \rightarrow \|b\rangle\|a\rangle$
SWAP		$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$	$\|00\rangle \rightarrow \|00\rangle$ $\|01\rangle \rightarrow \|10\rangle$ $\|10\rangle \rightarrow \|01\rangle$ $\|11\rangle \rightarrow \|11\rangle$

Quadro 11: Operador de troca.
Fonte: Elaborado pelo autor.

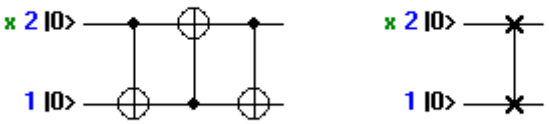


Figura 9: Circuito para a porta de troca.
Fonte: Adaptado de Nielsen e Chuang (2005).

2.4.6 Circuitos Quânticos

Da mesma forma que um circuito digital clássico, onde podem ser combinadas portas lógicas para produzir uma determinada configuração de bits de saída, na Computação Quântica são criados

circuitos quânticos, combinando-se um conjunto de portas quânticas para se alcançar um determinado estado desejado. A representação mais comum para um circuito quântico considera os q-bits como sendo "fios", representando o estado de uma partícula evoluindo no tempo. Na Figura 10 é mostrado um circuito quântico genérico ***U***, havendo um estado combinado de entrada $|q_1\rangle,|q_2\rangle...|q_n\rangle$, o processamento mediante um conjunto de portas quânticas, e o estado de saída resultante $|q'_1\rangle,|q'_2\rangle...|q'_n\rangle$. Os estados quânticos são manipulados utilizando-se as portas quânticas, de forma que apenas ao final é feita a medida para se saber os estados dos q-bits resultantes. O símbolo da medida é colocado ao final de cada fio representando o q-bit.

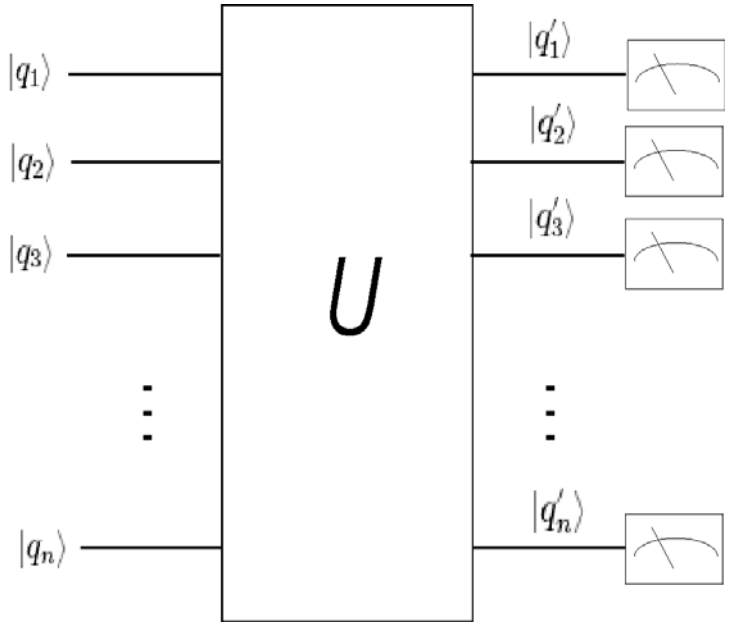


Figura 10: Representação genérica de um circuito quântico.
Fonte: Adaptado de Portugal et al (2004).

Pode-se representar o circuito quântico *U* da figura como sendo um operador que mapeia o conjunto de q-bits $|q_1\rangle,|q_2\rangle...|q_n\rangle$ para $|q'_1\rangle,|q'_2\rangle...|q'_n\rangle$, da seguinte forma,

$$U: \ |q_1\rangle|q_2\rangle...|q_n\rangle \rightarrow \ |q'_1\rangle|q'_2\rangle...|q'_n\rangle$$

Ao contrário dos circuitos lógicos clássicos, um circuito quântico não permite **realimentação** ou ***feedback***. Enquanto que um circuito digital pode ter diferentes direções e sentidos de propagação de um sinal, o estado dos q-bits em um circuito quântico evolui **da esquerda para a direita**. Uma característica importante dos circuitos quânticos, inexistente nos circuitos clássicos, é a reversibilidade. Um estado quântico final pode ter sua ordem de evolução invertida no tempo, sendo aplicado à saída das portas quânticas, e o estado inicial pode ser, portanto, recuperado. Este conceito está de acordo com a Mecânica Quântica, através da equação de Schröedinger, onde o hamiltoniano, que é a função de evolução unitária no tempo aplicada sobre um estado quântico, sempre existe e possui inversa (AHARONOV, 1998).

2.4.7 Produto Tensorial e Registradores de Q-bits

Quando são necessários mais q-bits em um circuito, eles podem ser combinados a partir de uma operação denominada **produto tensorial**, compondo desta forma um *registrador* de q-bits, similar aos registradores dos circuitos clássicos (AHARONOV, 1998). Um registrador de *n* q-bits pode ser construído a partir do produto tensorial entre eles:

$$|q_1\rangle \otimes |q_2\rangle \otimes ... \otimes |q_n\rangle$$

Ou, na forma compacta,

$$|q_1 q_2 ... q_n\rangle$$

Por exemplo, o produto entre três q-bits pode ser representado por

$$|0\rangle \otimes |0\rangle \otimes |0\rangle$$
$$|0\rangle \otimes |0\rangle \otimes |1\rangle$$
$$|0\rangle \otimes |1\rangle \otimes |0\rangle$$
$$|0\rangle \otimes |1\rangle \otimes |1\rangle$$
$$\ldots$$
$$|1\rangle \otimes |1\rangle \otimes |1\rangle$$

Na forma compacta,

$$|000\rangle$$
$$|001\rangle$$
$$|010\rangle$$
$$|011\rangle$$
$$\ldots$$
$$|111\rangle$$

ou ainda, utilizando a **notação decimal** para registradores quânticos (EKERT et al, 2000)

$$|0\rangle$$
$$|1\rangle$$
$$|2\rangle$$
$$|3\rangle$$
$$\ldots$$
$$|7\rangle$$

Deve-se notar que, na última forma, a representação do registrador de três q-bits $|0\rangle$ e $|1\rangle$ são diferentes dos estados quânticos de um q-bit $|0\rangle$ e $|1\rangle$. Na Figura 11 está representado um circuito quântico que produz os estados de Bell ou estados EPR (NIELSEN e CHUANG, 2005). Este circuito é composto de uma porta Walsh-Hadamard e uma porta C-NOT. De acordo com os estados dos q-bits de entrada, os diferentes estados emaranhados de dois q-bits são produzidos. A letra "x" indica o registrador quântico "x", os números são os índices dos q-bits e os estados $|0\rangle$ são os estados de entrada. No Quadro 12 são

mostrados os circuitos quânticos que geram os estados de Bell respectivos.

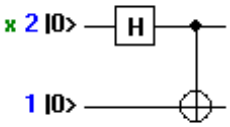
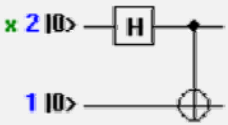
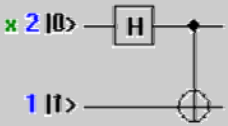
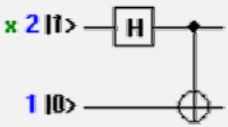
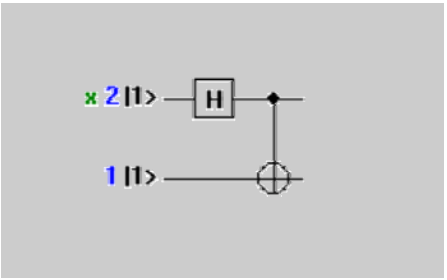


Figura 11: Circuito quântico para produzir estados de Bell.
Fonte: Adaptado de Nielsen e Chuang (2005).

Circuito	Estado de Bell
	$\beta_{00} = \frac{\|00\rangle + \|11\rangle}{\sqrt{2}}$
	$\beta_{01} = \frac{\|01\rangle + \|10\rangle}{\sqrt{2}}$
	$\beta_{10} = \frac{\|00\rangle - \|11\rangle}{\sqrt{2}}$



$$\beta_{11} = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Quadro 12: Circuitos possíveis para os estados de Bell.
Fonte: Adaptado de Nielsen e Chuang (2005).

Tomando-se o primeiro circuito com os q-bits de entrada $|00\rangle$, a Porta Walsh-Hadamard atuando sobre o segundo q-bit produz uma superposição de estados da forma,

$$\frac{|00\rangle + |10\rangle}{\sqrt{2}}$$

Ou seja, faz com que os q-bits assumam os dois estados $|00\rangle$ e $|10\rangle$ ao mesmo tempo. Deve-se notar que o primeiro q-bit mantém-se inalterado. A seguir, a porta CNOT modifica o primeiro q-bit (alvo), de forma que, quando o segundo q-bit (controle) assume o valor $|1\rangle$, o primeiro q-bit será invertido. Portanto, para o estado $|00\rangle$ não há alteração, mas o estado $|10\rangle$ torna-se $|11\rangle$. Portanto, a superposição torna-se:

$$\beta_{00} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

Para a representação matricial desta operação de produção dos estados de Bell, o estado de entrada deve ser visualizado como o produto tensorial dos dois q-bits, produzindo um vetor de 4 linhas,

$$|00\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}.$$

A operação Walsh-Hadamard atua apenas sobre o segundo q-bit. Considerando, porém, o vetor de estado dos dois q-bits de tamanho 4x1, o operador a ser aplicado a este vetor deve ter dimensões 4x4. Esta operação efetua a seguinte operação de multiplicação matricial 2x2 sobre um q-bit

$$H = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}$$

Deve-se então obter uma matriz 4x4 que faça a operação sobre os dois q-bits combinados. O primeiro q-bit não é alterado, de forma que é similar a uma multiplicação pela matriz identidade *I*. Da mesma forma que o produto tensorial entre dois estados quânticos gera um vetor 4x1, o operador 4x4 consiste, portanto, no produto tensorial

$$H \otimes I = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix} \otimes I = \frac{1}{\sqrt{2}}\begin{bmatrix} 1.I & 1.I \\ 1.I & -1.I \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

Então, o produto entre o vetor $|00\rangle$ por esta matriz $(H \otimes I)$ é,

$$(H \otimes I)\,|\,00\rangle = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

Por fim, a aplicação da operação CNOT (seção 2.4.5) é vista como a seguinte multiplicação matricial,

$$CNOT((H \otimes I)\,|\,00\rangle) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix} \cdot \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \\ 1 \\ 0 \end{bmatrix} = \frac{1}{\sqrt{2}}\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}$$

sendo tal vetor 4x1 resultante o estado de Bell β_{00}.

2.4.8 Superposição e Interferência

A porta Walsh-Hadamard pode ser utilizada para gerar estados superpostos de mais q-bits. Por exemplo, o circuito da Figura 12 gera uma superposição de oito estados para três q-bits.

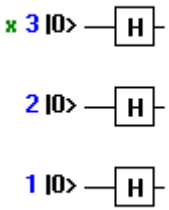


Figura 12: Circuito com 3 portas Walsh-Hadamard para gerar 8 estados superpostos.
Fonte: Elaborado pelo autor.

A partir da entrada dos três q-bits $|\psi\rangle=|000\rangle$, as portas Walsh-Hadamard aplicadas irão gerar o seguinte estado superposto $|\psi'\rangle$ (EKERT et al, 2000),

$$|\psi'\rangle = \frac{1}{2\sqrt{2}}\left(|000\rangle+|001\rangle+|010\rangle+|011\rangle+|100\rangle+|101\rangle+|110\rangle+|111\rangle\right)$$

Ou ainda

$$|\psi'\rangle = \frac{1}{2\sqrt{2}}\left(|0\rangle+|1\rangle+|2\rangle+|3\rangle+|4\rangle+|5\rangle+|6\rangle+|7\rangle\right)$$

Ou, em forma de somatório,

$$|\psi'\rangle = \frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle$$

A superposição de estados é um dos recursos que torna a Computação Quântica peculiar, sendo a base para muitos algoritmos

quânticos. Os estados superpostos indicam que o q-bit assume, **ao mesmo tempo**, todos os estados que fazem parte da superposição. Quando tal estado é processado pelas próximas portas quânticas do circuito, todos os estados que fazem parte são processados simultaneamente, mantendo-se a superposição (AHARONOV, 1998, p.6). O estado superposto cessa somente quando se faz a medida do estado quântico. Assim, a porta Walsh-Hadamard faz com que todos os estados resultantes tenham igual probabilidade de obtenção após a medida. Com relação ao exemplo anterior, quando se mede o estado $|\psi'\rangle$, apenas um dentre os oito estados será retornado, com probabilidade $1/8 = |1/2\sqrt{2}|^2$.

Portanto, para *n* q-bits, com o resultado da superposição sendo expandido para

$$|\psi'\rangle = \frac{1}{\sqrt{2^n}} \sum_{i=0}^{n-1} |i\rangle ,$$

pode-se verificar o poder da Computação Quântica em trabalhar de forma exponencial com todos os 2^n estados possíveis simultaneamente. O problema que se coloca, a partir daí, é como extrair tal informação exponencial do sistema quântico (AHARONOV, 1998). Quando se observa ou se faz a medição do sistema, acontece o colapso da função de onda. Este colapso faz com que apenas um dos estados manipulados anteriormente em forma de superposição seja recuperável, e o montante de informação exponencial termina por se perder. Conforme Aharonov (1998), para obter vantagem do paralelismo exponencial, deve-se combiná-lo com outro aspecto quântico, a **interferência**. A interferência permite que muitas computações sejam feitas em paralelo, exponencialmente, havendo cancelamento tal como o que acontece devido à interferência destrutiva de ondas. A meta, portanto, é combinar este cancelamento de forma que apenas aquelas computações que são de interesse permaneçam, e todas as restantes sejam descartadas (AHARONOV, 1998). Isto pode ser visualizado no funcionamento do algoritmo quântico de busca de Grover, por exemplo.

2.4.9 Paralelismo Quântico

Através do paralelismo quântico, um circuito quântico pode fazer a **avaliação de uma função** para vários estados quânticos simultaneamente, em conjunto com o princípio da superposição. David

Deutsch (1985) propôs um circuito na forma de uma "caixa-preta" fazendo uso do paralelismo quântico, permitindo saber se uma função $f(x) \in \{0,1\}$ apresentaria uma característica a partir de apenas uma única avaliação de $f(x)$. Esta característica era determinar se $f(x)$ é uma função **constante** (tendo o mesmo valor, ou 0 ou 1, para qualquer entrada x) ou **balanceada** (fornecendo 0's e 1's para diferentes entradas x). A Figura 13 mostra o circuito quântico para o algoritmo de Deustch. O circuito contém dois q-bits, $|x\rangle$ e $|y\rangle$. A transformação relativa à função $f(x)$ (denominada de U_f) atuando sobre os dois q-bits é da forma

$$U_f : \ |x\rangle|y\rangle \rightarrow \ |x\rangle|y \ \oplus f(x)\rangle$$

O estado inicial $|\psi_0\rangle$ do circuito é dado por

$$|\psi_0\rangle = |0\rangle|1\rangle$$

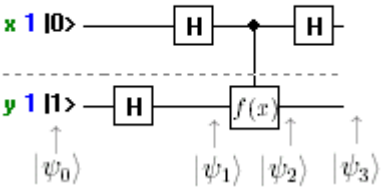


Figura 13: Representação do algoritmo de Deutsch. A função $f(x)$ representa a transformação U_f.
Fonte: Adaptado de Nielsen e Chuang (2005).

Após a operação Walsh-Hadamard sobre o q-bit $|y\rangle$, tem-se

$$|\psi_1\rangle = |x\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

Assim, aplicando-se U_f ao estado $|\psi_1\rangle$, obtém-se o estado $|\psi_2\rangle$

$$|\psi_2\rangle = U_f\,|\psi_1\rangle = |x\rangle\left[\frac{|f(x)\rangle - |1 \oplus f(x)\rangle}{\sqrt{2}}\right]$$

A expressão do segundo q-bit pode ser escrita de forma equivalente a

$$|\psi_2\rangle = |x\rangle(-1)^{f(x)}\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

Com a aplicação da porta Walsh-Hadamard atuando-se sobre o q-bit $|x\rangle$ (na verdade, havendo atuado já no estado $|\psi_1\rangle$) , tem-se

$$|\psi_2\rangle = \frac{1}{\sqrt{2}}\left[|0\rangle(-1)^{f(0)} + |1\rangle(-1)^{f(1)}\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right]$$

A partir da função $f(x)$, o estado $|\psi_2\rangle$ pode apresentar duas situações

$$|\psi_2\rangle = \begin{cases} \pm\frac{1}{\sqrt{2}}\left[|0\rangle + |1\rangle\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], & \text{se } f(0) = f(1) \\ \pm\frac{1}{\sqrt{2}}\left[|0\rangle - |1\rangle\right]\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], & \text{se } f(0) \neq f(1) \end{cases}$$

Após a última porta Walsh-Hadamard sobre o primeiro q-bit, tem-se o estado $|\psi_3\rangle$

$$|\psi_3\rangle = \begin{cases} \pm|\mathbf{0}\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], & \text{se } f(0) = f(1) \\ \pm|\mathbf{1}\rangle\left[\frac{|0\rangle - |1\rangle}{\sqrt{2}}\right], & \text{se } f(0) \neq f(1) \end{cases}$$

Medindo-se o q-bit $|x\rangle$ apenas uma vez, pode-se determinar a propriedade da função, sendo **constante** (obtendo-se $|0\rangle$ com resultado da medida) ou **balanceada** (obtendo-se $|1\rangle$ após a medida). Portanto, o uso do paralelismo, em conjunto com a superposição quântica, permitiu saber-se a característica da função avaliando-se a transformação U_f no

circuito quântico de Deutsch **uma única vez**. Para o algoritmo clássico, duas vezes seriam necessárias, para se avaliar o tipo da função.

O algoritmo de Deutsch-Josza (1992) estende o problema da avaliação da função de um q-bit para *n* q-bits de entrada, permitindo novamente a avaliação uma **única vez para vários q-bits**, sendo que o caso clássico exigiria $O(N)$ avaliações da função $f(x)$ para saber se é constante ou balanceada (NIELSEN e CHUANG, 2005; EKERT et al, 2000; PRESKILL, 1995). Na Figura 14 está um exemplo do algoritmo Deutsch-Josza, utilizando três q-bits para o primeiro registrador. Note que o tamanho do segundo registrador não muda, sendo ainda de um q-bit.

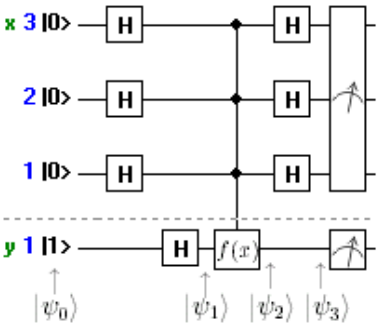


Figura 14: Circuito simulador do algoritmo de Deutsch-Josza com três q-bits.
Fonte: Adaptado de Nielsen e Chuang (2005).

O uso de transformações na forma de "caixas-pretas", tal como a função U_f, são também denominadas de **oráculos**, utilizados amplamente nos algoritmos quânticos.

2.4.10 Emaranhamento

Consistindo numa das propriedades mais estranhas da Mecânica Quântica, o emaranhamento começou a adquirir importância a partir do trabalho de Einstein, Podolsky e Rosen (1935) numa tentativa de demonstrar a **incompletude** da Mecânica Quântica na explicação dos fenômenos quânticos. O estudo do fenômeno EPR foi mais tarde retomado por John Bell (1964), que formulou a **desigualdade de Bell**,

também conhecida como **desigualdade CHSH** (termo devido aos seus autores que a formularam de forma independente, Clauser, Horne, Shimony e Holt). Esta desigualdade deveria ser violada, caso a Mecânica Quântica fosse válida (NIELSEN e CHUANG, 2005). Um dos testes experimentais mais relevantes, executados por Aspect et al (1982) mostraram esta violação, validando a Mecânica Quântica e demonstrando que o senso comum era errôneo na descrição dos fenômenos quânticos.

Dois objetos quânticos estão **emaranhados** quando a alteração de uma grandeza, (como, por exemplo, o spin) em um dos objetos afeta instantaneamente o outro, não importa a distância que estejam entre si. Utilizando-se um dos estados de Bell como exemplo, onde dois q-bits estão emaranhados:

$$\beta_{11} = \frac{|01\rangle - |10\rangle}{\sqrt{2}}$$

Este estado equivale à representação em forma de produto tensorial

$$\beta_{11} = \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}$$

Lembrando-se que o circuito quântico que produz este estado está representado na Figura 15.

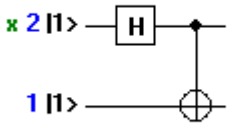


Figura 15: Circuito quântico para obter o estado de Bell β_{11}.
Fonte: Adaptado de Nielsen e Chuang (2005).

A representação de tal estado é bem estudada na teoria quântica pela denominação de "**singleto de Bohm**" (PESSOA JÚNIOR, 2006; ZEILINGER, 2005). Este estado surge, por exemplo, do decaimento de

uma partícula que possui spin 0 (zero) em duas outras partículas, viajando em sentidos diferentes ao longo de um eixo qualquer (Figura 16). Cada partícula possui agora spin ½, seguindo em direções contrárias uma da outra. A função de onda que representa este estado é *anti-simétrica* (devido ao sinal de menos, pois são partículas **fermiônicas**). A partícula da esquerda possui spin "para cima", enquanto que a da direita possui spin "para baixo". Enquanto não se mede o sistema, não há como saber o spin de cada partícula**. Mas quando o spin de uma é medido, instantaneamente sabe-se da orientação do spin da outra (em sentido contrário), não importando a distância presente entre elas**.

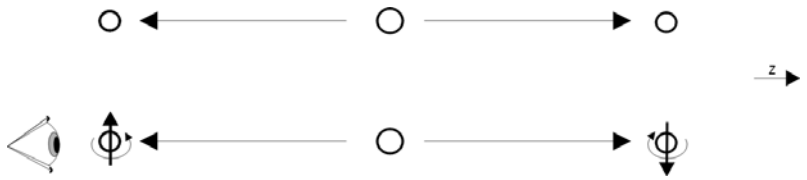


Figura 16: Ilustração do estado emaranhado referente ao singleto de Bohm.
Fonte: Adaptado de Zeilinger (2005).

Um estado emaranhado não permite a sua decomposição em um produto tensorial. Por exemplo, o estado a seguir de dois q-bits pode ser decomposto em produto tensorial

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \mid 00 \rangle .$$

Entretanto, os estados emaranhados

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix} \text{ ou } \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

não permitem a sua representação utilizando q-bits simples em um produto tensorial.

O emaranhamento permite o uso de uma forma peculiar de comunicação, a **codificação superdensa** (PRESKILL, 1998; STEANE, 1997; NIELSEN e CHUANG, 2005). Por meio de um canal quântico, pode-se transmitir mais bits de comunicação do que em um canal clássico. Supondo-se que a fonte e o destino (denominados de "Alice" e "Bob"), estejam a uma grande distância, um do outro; o objetivo é Alice transmitir informação clássica para Bob na forma de bits clássicos, dispondo de apenas um q-bit. Caso Alice e Bob compartilhem um par de q-bits no estado emaranhado de Bell

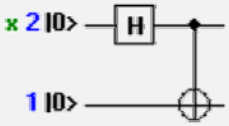
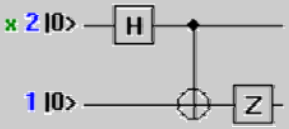
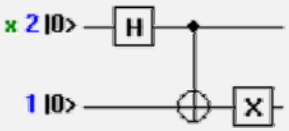
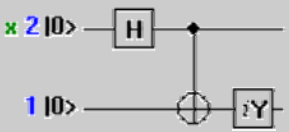
$$\beta_{00} = \frac{|00\rangle + |11\rangle}{\sqrt{2}}$$

de forma que Alice possui um q-bit, e Bob o outro q-bit. Para que Alice envie dois bits de comunicação clássica para Bob, ela deve enviar o seu q-bit para Bob. Quatro situações podem ser obtidas com esta situação:

1) Se Alice deseja enviar a seqüência de bits "00" para Bob, ela não precisa fazer nada com o seu q-bit, enviando do jeito que está.
2) Se Alice deseja enviar a seqüência de bits "01" para Bob, antes de enviar o seu q-bit, ela deve aplicar a porta $\boldsymbol{Z}$ ao seu q-bit.
3) Se Alice deseja enviar a seqüência "10" para Bob, deve aplicar a porta $\boldsymbol{X}$ ao seu q-bit.
4) Se Alice deseja enviar a seqüência "11" para Bob, deve aplicar a porta $i\boldsymbol{Y}$ ao seu q-bit.

Portanto, os dois q-bits mapearão cada um dos estados de Bell, de acordo com o Quadro 13. Neste quadro estão os circuitos quânticos que simulam os estados após a ação de Alice sobre o seu q-bit. Os estados de Bell podem ser medidos depois por Bob para determinar qual a seqüência de bits Alice quis enviar. Desta forma, dois bits clássicos podem ser comunicados através de um canal quântico de um q-bit (NIELSEN e CHUANG, 2005).

Nagy e Akl (2005) comentam que o emaranhamento não deve ser visto como uma curiosidade quântica, visto que é um **recurso físico** que pode ser utilizado para a resolução de problemas a partir de novos caminhos. Consideram até mesmo como sendo **crucial** para o próprio futuro da Computação Quântica e Informação Quântica.

Seqüência de bits	Estado de Bell	Circuito Quântico Simulador da ação de Alice
00	$\beta_{00} = \frac{\|00\rangle + \|11\rangle}{\sqrt{2}}$	
01	$\beta_{10} = \frac{\|00\rangle - \|11\rangle}{\sqrt{2}}$	
10	$\beta_{01} = \frac{\|01\rangle + \|10\rangle}{\sqrt{2}}$	
11	$\beta_{11} = \frac{\|01\rangle - \|10\rangle}{\sqrt{2}}$	

Quadro 13: Esquema para codificação superdensa.
Fonte: Adaptado de Nielsen e Chuang (2005).

2.4.11 Medida

Enquanto não se observa a evolução de um sistema quântico, e este sistema evolui num regime fechado em termos de interação com o ambiente, o comportamento desta evolução é **determinístico** e é regido pela equação de Schröedinger, mediante transformações unitárias. Isto está de acordo com o postulado da evolução unitária proveniente da Mecânica Quântica (NIELSEN e CHUANG, 2005). Portas quânticas podem ser descritas como as transformações unitárias sobre um estado quântico, sendo a evolução, dentro das condições fechadas, previsível. Porém, é necessário efetuar observações ou medidas sobre o sistema quântico, e isto causa o **colapso** do sistema. A medida em um sistema quântico manifesta um caráter **probabilístico**, trazendo a imprevisibilidade ao sistema. A Mecânica Quântica apenas traz uma descrição das probabilidades de se obter um ou outro estado quântico, sendo as causas que motivam a escolha de estados, após a medida, desconhecidas.

De acordo com o postulado da medida, as medidas quânticas podem ser descritas mediante operadores de medida M_m. Tais operadores atuam sobre o espaço de estados de um sistema quântico, com o índice m se referindo aos possíveis resultados da medida (NIELSEN e CHUANG, 2005). Por exemplo, supõe-se um estado $|\psi\rangle$ qualquer sendo um conjunto de dois estados superpostos,

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle$$

com as amplitudes α_0 e $\alpha_1 \in$ X, e satisfazendo a condição de normalização $|\alpha_0|^2+|\alpha_1|^2=1$. Os operadores de medida são, portanto, M_0 e M_1, dados por

$$M_0 = \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} \text{ e } M_1 = \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}.$$

A probabilidade de que uma medida possa ser obtida é resultante da multiplicação do operador de medida com o estado pelo seu adjunto correspondente (ou através da representação dual do vetor de estados)

$$p_m = (M_m|\psi\rangle)^+(M_m|\psi\rangle) = \langle\psi|M_m^+M_m|\psi\rangle .$$

Com o resultado após a medida sendo normalizado,

$$\frac{M_m \mid \psi\rangle}{\sqrt{p_m}}.$$

Para o exemplo, a probabilidade de se obter o estado $|0\rangle$ após a medida é

$$p_0 = (M_0 \mid \psi\rangle)^+ (M_0 \mid \psi\rangle) = \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} . \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \right)^+ . \left(\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} . \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \right) =$$

$$= \begin{bmatrix} \alpha_0 & 0 \end{bmatrix} . \begin{bmatrix} \alpha_0 \\ 0 \end{bmatrix} = \mid \alpha_0 \mid^2$$

O estado final após a medida é

$$\frac{\alpha_0 \mid 0\rangle}{\sqrt{\mid \alpha_0 \mid^2}} = \mid 0\rangle .$$

Da mesma forma, a probabilidade de se obter o estado $|1\rangle$ após a medida vem a ser

$$p_1 = (M_1 \mid \psi\rangle)^+ (M_1 \mid \psi\rangle) = \left(\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \right)^+ . \left(\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} . \begin{bmatrix} \alpha_0 \\ \alpha_1 \end{bmatrix} \right) =$$

$$= \begin{bmatrix} 0 & \alpha_1 \end{bmatrix} . \begin{bmatrix} 0 \\ \alpha_1 \end{bmatrix} = \mid \alpha_1 \mid^2$$

com o estado final, neste caso, sendo

$$\frac{\alpha_1 \mid 1\rangle}{\sqrt{\mid \alpha_1 \mid^2}} = \mid 1\rangle .$$

Figura 17: Símbolo do operador de medida para um circuito quântico. Fonte: Adaptado de Nielsen e Chuang (2005).

A medição quântica pode ainda se diferenciar com relação à forma com que é feita. Existem as medições projetivas, quando se trabalha com estatística de medidas, com várias medidas sendo efetuadas sobre um observável e calculando-se valores médios para as medidas. Porém, existem casos onde o estado do sistema após a medição é de pouco interesse, sendo importante identificar a probabilidade dos diferentes resultados que podem acontecer após a medida. A medição é feita apenas uma vez e o experimento é então concluído. Tal caso utiliza o *formalismo POVM*, do inglês, *Positive Operator-Valued Measure* (NIELSEN e CHUANG, 2005). Para o operador de medição, utiliza-se a representação da Figura 17.

2.4.12 Descoerência

A descoerência (também chamada de "decoerência") é um dos problemas principais que estão relacionados à estabilidade de um sistema quântico e a interpretação fidedigna de resultados de medidas. Para que a evolução unitária de estados quânticos em um sistema aconteça de forma controlada, o sistema deve ser **isolado** do seu ambiente. Porém, é virtualmente impossível o isolamento perfeito de um grande sistema quântico do seu ambiente. As interações entre um dispositivo quântico e seu ambiente estabelecem correlações não-locais entre os dois (PRESKILL, 1998). Mas um sistema quântico deve ser medido, com a medida trazendo a informação do mundo quântico para o mundo clássico. A interpretação da descoerência está no cerne da questão referente ao limite de onde termina o mundo quântico e começa o clássico (ZUREK, 1991; ZUREK, 2002). Zeilinger (2005) ressalta o papel da informação no processo de descoerência, pois o fenômeno da interferência quântica, a partir da superposição de estados, só se manifesta se não está à disposição, em parte alguma, qualquer informação sobre qual estado o objeto quântico assumiu.

Um sistema tal como um computador quântico precisa então ser medido ao final de seu processamento, de forma que o aparelho utilizado na medição também se correlaciona com o sistema quântico, aumentando assim o sistema total. A observação de um sistema em superposição causaria o colapso do mesmo, e os estados subseqüentes à medida seriam individualmente e aleatoriamente os estados ortogonais da base. Entretanto, existem sistemas quânticos (tais como os dispositivos que utilizam RMN, com interações quânticas acontecendo com o conjunto total das moléculas que fazem parte da amostra) contendo um grande número de indivíduos, sendo denominados de

coletivos (***ensemble***) **estatísticos**. O formalismo para tratar estes coletivos vão além daquele considerado para estados individuais como o de vetor de estados quânticos, sendo melhor abordados através dos operadores **densidade** ou **matriz densidade** (NIELSEN e CHUANG, 2004). Enquanto que para sistemas com poucos elementos quânticos interagindo, a abordagem do colapso da função de onda é aplicada, para coletivos estatísticos a descoerência se aplica melhor, por esta ser um processo estatístico (PESSOA JÚNIOR, 2006).

2.5 Algoritmos Quânticos

A ideia central por trás do desenvolvimento de algoritmos quânticos está em obter, a partir de um novo paradigma de computação, resultados melhores e mais eficientes do que a abordagem tradicional dos chamados algoritmos clássicos. Em termos de complexidade computacional, com os algoritmos clássicos existe a dificuldade de se obter soluções de uma série de problemas em tempo polinomial, os chamados **NP-completos**. Tais algoritmos permitem a resolução dos problemas, porém necessitando de tempo exponencial ou fatorial. A esperança era a de que os algoritmos quânticos pudessem proporcionar, pelo uso das características de processamento exponencial e paralelismo, as soluções em tempo polinomial dos problemas **NP**-completos.

De acordo com Nielsen e Chuang (2005), as duas classes de problemas mais importantes são denominadas **P** e **NP.** A classe **P** é considerada a classe dos problemas que podem ser resolvidos de forma rápida em um computador clássico. A classe **NP** (*Non-deterministic Polynomial*), por sua vez, se refere à classe de problemas onde as *soluções* podem ser verificadas eficientemente em um computador clássico. Se um problema está em **NP**, não é requerido que exista um algoritmo de tempo polinomial que possa resolvê-lo. É necessária a existência de um algoritmo que possa testar uma solução e verificar se é correta em tempo polinomial (AHARONOV, 1998). Cormen et al (2002) assinalam que qualquer problema em **P** está em **NP**, pois pode ser resolvido em tempo polinomial sem necessidade de verificação. Porém, acredita-se que **P** seja diferente de **NP**, devido à existência dos problemas **NP**-completos. Os problemas **NP**-completos seriam considerados os problemas mais difíceis dentro da classe **NP**, ao qual se reduziriam os problemas **NP**. Se qualquer problema **NP**-completo pudesse ser resolvido em tempo polinomial, então todo problema em **NP** teria uma solução em tempo polinomial, ou seja, **P**=**NP**. Entretanto, não

foi descoberto ainda nenhum algoritmo de tempo polinomial para resolver problemas **NP**-completos (CORMEN et al, 2002).

Outro aspecto relevante a ser considerado é o caráter determinístico na obtenção de soluções dos problemas. Preskill (1998) comenta inclusive da possibilidade de um computador clássico simular um circuito quântico; entretanto, a necessidade de armazenamento cresce exponencialmente com o aumento de q-bits no circuito. Mas se a condição de determinismo for relaxada, aceitando-se soluções probabilísiticas com alguma margem de erro, o desempenho dos algoritmos quânticos é superior aos clássicos em certa classe de problemas.

Nielsen e Chuang (2005) colocam que certos tipos de problemas podem ser resolvidos utilizando-se algoritmos aleatórios que o fazem em tempo polinomial, aceitando-se uma probabilidade de erro limitada. Esta classe de problemas é denominada **BPP** (*Bounded Probability Polynomial*). Nesta classe podem ser enquadrados os algoritmos genéticos ou redes neurais artificiais, por exemplo. Aharonov (1998) coloca que a margem de aceitação de uma solução de um problema na classe **BPP** seria acima de 2/3.

Existe a contrapartida da classe **BPP** na computação quântica, a **BQP** (*Bounded Quantum Polynomial*), que é definida como a classe de algoritmos quânticos que resolvem os problemas de forma eficiente em um computador quântico, aceitando-se uma margem de erro (NIELSEN e CHUANG, 2005).

Entretanto, não se sabe ainda qual a relação entre a classe **BQP** e as classes **P** e **NP**. Os computadores quânticos podem resolver problemas de **P** com eficiência, mas existiriam problemas fora de **P** que eles não poderiam resolver com a mesma eficiência. O que está bem claro, conforme apontado por Nielsen e Chuang (2005) é que "a teoria da computação quântica apresenta interessantes e significativos desafios às noções tradicionais da computação". E isto é importante, na medida em que tal modelo teórico da Computação Quântica é *experimentalmente* realizável, pois a teoria está de concordância com a forma como a Natureza funciona. Não fosse este o caso, "a computação quântica seria uma mera curiosidade matemática" (NIELSEN e CHUANG, 2005).

Com o intuito de aproveitar as propriedades de superposição e emaranhamento, permitindo por sua vez uma eficiência maior em relação a algoritmos clássicos em termos de complexidade, alguns algoritmos quânticos foram propostos. A combinação do paralelismo

exponencial e a interferência (proporcionada pelo fenômeno da superposição) torna a Computação Quântica mais poderosa e possui uma regra importante nos algoritmos quânticos (AHARONOV, 1998). O algoritmo de Deutsch mostrado na seção 2.4.9 ilustra o uso da superposição para permitir, através de uma única avaliação, a característica desta função, utilizando os dois valores de avaliação da função $f(x)$ simultaneamente.

Porém, apesar dos esforços, a quantidade de algoritmos quânticos desenvolvidos até agora permanecem poucos (STEANE, 1997). De acordo com Nielsen e Chuang (2005) os algoritmos quânticos existem organizados sob a forma de três classes distintas:

1) Algoritmos Quânticos de Busca;
2) Algoritmos Quânticos baseados na Transformada de Fourier;
3) Simulações Quânticas.

Shor (2005) propõe uma divisão mais recente em quatro classes, adicionando às três citadas anteriormente a classe referente à Computação Quântica Adiabática.

2.5.1 Algoritmo Quântico de Busca de Grover

Grover (1996) propôs um algoritmo de busca que permite encontrar um elemento em um espaço de busca contendo N elementos, **sem nenhum conhecimento prévio** sobre a estrutura da informação contida nele, com eficiência $O(\sqrt{N})$. Os algoritmos clássicos de busca apresentam complexidade $O(N)$; portanto, o algoritmo de Grover apresenta ganho quadrático de velocidade neste tipo de busca. Este ganho quadrático pode ser utilizado na aceleração das soluções de problemas **NP**-completos (NIELSEN e CHUANG, 2005).

A busca mediante o algoritmo de Grover é feita de forma a aumentar a probabilidade de se encontrar o elemento desejado dentro do espaço de q-bits. O algoritmo altera, portanto, as amplitudes de todas as possíveis soluções através da aplicação de operadores específicos. Este processo é denominado de **amplificação de amplitude quântica** (BRASSARD et al, 1998).

O algoritmo de Grover opera sobre um registrador de n q-bits, iniciando no estado $|00...0\rangle$. Da mesma forma que o algoritmo de Deutsch, existe um segundo q-bit auxiliar, no estado $(|0\rangle - |1\rangle)/\sqrt{2}$,

$$|\psi_0\rangle = |00...0\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right].$$

Baseado em Nielsen e Chuang (2005), a seguir são listadas as três operações necessárias para se efetuar a busca com o algoritmo de Grover:

1) **Operação de transformação de Walsh-Hadamard:** que coloca todos os elementos possíveis de busca {0..N-1} em superposição:

$$|\psi_1\rangle = \frac{1}{N^{1/2}} \sum_{x=0}^{N-1} |x\rangle$$

2) **Aplicação do oráculo**: o oráculo efetua a seguinte transformação sobre o estado $|\psi_1\rangle$

$$U_f : \ |x\rangle |y\rangle \rightarrow \ |x\rangle |y \ \oplus f(x)\rangle$$

A função $f(x)$ retorna 1 caso o elemento buscado seja encontrado, senão retorna 0. O oráculo pode ser representado também como

$$U_f : \ |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right] \rightarrow \ (-1)^{f(x)} |x\rangle \left[\frac{|0\rangle - |1\rangle}{\sqrt{2}} \right]$$

Como o q-bit auxiliar não muda, convenciona-se o uso apenas do primeiro q-bit,

$$U_f : \ |x\rangle \rightarrow \ (-1)^{f(x)} |x\rangle$$

O oráculo então "marca" a solução do problema de busca, mudando a sua fase.

3) **Aplicação de um deslocamento de fase:** feita no registrador quântico $|x\rangle$, antecedido e sucedido de transformações Walsh-Hadamard. O deslocamento de fase é feito em todos os estados da base computacional menos o estado $|0\rangle$. A operação de deslocamento é representada utilizando-se o produto externo para o estado $|0\rangle$ e a matriz identidade, da forma $2|0\rangle\langle 0| - I$, de forma que os três operadores combinados,

$$|x\rangle \rightarrow H(2|0\rangle\langle 0|-I)H|x\rangle$$

podem ser representados pelo operador único $2|\psi\rangle\langle\psi|-I$.

A *iteração de Grover*, é representada pela combinação dos passos 2 e 3, sendo representada por

$$G = (2|\psi\rangle\langle\psi|-I)U_f$$

e deve ser aplicada iterativamente para aumentar a probabilidade de obtenção do elemento desejado, ao se efetuar a medida do registrador $|x\rangle$. Geometricamente, a iteração de Grover perfaz reflexões do vetor em relação ao vetor inicial, resultando numa rotação no espaço de busca (NIELSEN e CHUANG, 2005).

Na sequência é ilustrado um exemplo para explicar o funcionamento do algoritmo de Grover, para um circuito quântico de três q-bits, o número de estados possíveis será $N=2^3=8$. O conjunto de estados é, portanto, ($|0\rangle$, $|1\rangle$, $|2\rangle$, $|3\rangle$, $|4\rangle$, $|5\rangle$, $|6\rangle$, $|7\rangle$). Para se buscar o estado $|6\rangle$ no conjunto, será executado o algoritmo de Grover, começando-se com o estado inicial e a aplicação das portas Walsh-Hadamard aos três q-bits. O estado inicial começa então com (será ignorado, para fins de explicação do exemplo, o q-bit auxiliar)

$$|\psi_0\rangle = |000\rangle = |\mathbf{0}\rangle$$

Com a aplicação das portas Walsh-Hadamard, têm-se a superposição dos estados dos q-bits,

$$|\psi_1\rangle = \frac{1}{2\sqrt{2}}\sum_{x=0}^{7}|x\rangle = \frac{1}{2\sqrt{2}}\left(|0\rangle+|1\rangle+|2\rangle+|3\rangle+|4\rangle+|5\rangle+|6\rangle+|7\rangle\right)$$

O próximo passo será aplicar o oráculo a esta superposição. O oráculo deverá, então, marcar o estado a ser encontrado mudando o sinal da amplitude respectiva ao estado. A função $f(x)=0$ para todos os estados, exceto $f(6)=1$:

$$|\psi_2\rangle = \frac{1}{2\sqrt{2}}\sum_{x=0}^{7}-(-1)^{f(x)}|x\rangle = \frac{1}{2\sqrt{2}}\left(|0\rangle+|1\rangle+|2\rangle+|3\rangle+|4\rangle+|5\rangle-|\mathbf{6}\rangle+|7\rangle\right)$$

O passo seguinte é o deslocamento de fase, sendo aplicado o operador $2|\psi\rangle\langle\psi|-I$ ao estado obtido após a aplicação do oráculo. Construindo o operador de deslocamento de fase, fazendo-se o produto externo do estado ψ_1,

$$|\psi_1\rangle\langle\psi_1| = \frac{1}{2\sqrt{2}}\begin{bmatrix}1\\1\\1\\1\\1\\1\\1\\1\end{bmatrix}\cdot\frac{1}{2\sqrt{2}}\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix} =$$

$$= \frac{1}{8}\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix}$$

O operador de deslocamento de fase é, portanto,

$$2|\psi_1\rangle\langle\psi_1| - I = \frac{1}{4}\begin{bmatrix}1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & 1\end{bmatrix} - I = \frac{1}{4}\begin{bmatrix}-3 & 1 & 1 & 1 & 1 & 1 & 1 & 1\\1 & -3 & 1 & 1 & 1 & 1 & 1 & 1\\1 & 1 & -3 & 1 & 1 & 1 & 1 & 1\\1 & 1 & 1 & -3 & 1 & 1 & 1 & 1\\1 & 1 & 1 & 1 & -3 & 1 & 1 & 1\\1 & 1 & 1 & 1 & 1 & -3 & 1 & 1\\1 & 1 & 1 & 1 & 1 & 1 & -3 & 1\\1 & 1 & 1 & 1 & 1 & 1 & 1 & -3\end{bmatrix}$$

Aplicando-se o operador ao vetor ψ_2 produzido pela aplicação do oráculo, obtém-se

$$\psi_3 = (2|\psi_1\rangle\langle\psi_1| - I)\psi_2 = \frac{1}{4}\begin{bmatrix} -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -3 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -3 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -3 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & -3 \end{bmatrix} \cdot \frac{1}{2\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -1 \\ 1 \end{bmatrix} =$$

$$= \frac{1}{8\sqrt{2}}\begin{bmatrix} 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ 2 \\ \mathbf{10} \\ 2 \end{bmatrix} = \frac{1}{4\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ \mathbf{5} \\ 1 \end{bmatrix} = \frac{1}{4\sqrt{2}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle + \mathbf{5}|\mathbf{6}\rangle + |7\rangle\right)$$

As amplitudes de todos os estados foram reduzidas ($1/4\sqrt{2} \cong 0{,}177$), exceto a amplitude do estado procurado que foi aumentada ($5/4\sqrt{2} \cong 0{,}883$). Antes do deslocamento de fase, as probabilidades de obtenção dos estados eram iguais (a probabilidade é igual ao quadrado da amplitude, e inicialmente estava em 12,5% para todos os estados). Após, a probabilidade de obtenção do estado $|6\rangle$ foi amplificada para aproximadamente 77%, enquanto que para os estados restantes foi atenuada para aproximadamente 3%.

A iteração de Grover pode ser novamente aplicada, ou seja, o passo do oráculo mais o deslocamento de fase. Após o passo do oráculo, têm-se novamente, o estado procurado com o sinal invertido,

$$\psi_4 = U_f\psi_3 = \frac{1}{4\sqrt{2}}\left(|0\rangle + |1\rangle + |2\rangle + |3\rangle + |4\rangle + |5\rangle - \mathbf{5}|\mathbf{6}\rangle + |7\rangle\right)$$

E repetindo-se o passo do deslocamento de fase, tem-se o estado ψ_5

$$\psi_5 = (2|\psi_1\rangle\langle\psi_1| - I)\psi_4 = \frac{1}{4}\begin{bmatrix} -3 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -3 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & -3 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & -3 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & -3 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & -3 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & -3 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & -3 \end{bmatrix} \cdot \frac{1}{4\sqrt{2}}\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ -5 \\ 1 \end{bmatrix} =$$

$$= \frac{1}{16\sqrt{2}}\begin{bmatrix} -2 \\ -2 \\ -2 \\ -2 \\ -2 \\ -2 \\ \mathbf{22} \\ -2 \end{bmatrix} = \frac{1}{8\sqrt{2}}\begin{bmatrix} -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ -1 \\ \mathbf{11} \\ -1 \end{bmatrix} = \frac{1}{8\sqrt{2}}\left(-|0\rangle - |1\rangle - |2\rangle - |3\rangle - |4\rangle - |5\rangle + \mathbf{11}|\mathbf{6}\rangle - |7\rangle\right)$$

Agora, a probabilidade do estado procurado $|6\rangle$ aumentou para aproximadamente 94,5%, enquanto que as probabilidades dos estados restantes diminuíram para 0,7%. Efetuando-se a medida do registrador quântico, a probabilidade de se encontrar o estado procurado é bastante alta em relação aos outros.

Analisando-se a execução do algoritmo, foram necessárias duas iterações de Grover mais a aplicação da operação Walsh-Hadamard para encontrar o estado buscado. O algoritmo clássico iria proceder a busca seqüencial (lembrando-se ainda que o algoritmo de Grover atua sobre um conjunto de estados sem uma estrutura de ordenação) em seis iterações de comparação. A vantagem do algoritmo de Grover para os clássicos começa a ficar mais evidente para circuitos com mais q-bits, com a complexidade sendo reduzida em média de forma quadrática ($O(\sqrt{N})$) em relação aos algoritmos clássicos (em média, $O(N)$). As operações referentes ao algoritmo de Grover podem ser reduzidas a portas quânticas usuais (NIELSEN e CHUANG, 2005), tais como as descritas na seção 2.4.5.

2.5.2 Estimativa de Fase

De acordo com Nielsen e Chuang (2004), a estimativa de fase não deve ser considerada por si só como um algoritmo, e sim como uma espécie de sub-rotina, sendo interessante quando é combinada com outros módulos. A estimativa de fase permite que se obtenha **a fase de um estado quântico** com certa margem de erro. O procedimento para a estimativa de fase está na base do procedimento de busca de ordem e fatoração do algoritmo de Shor, e é a base para o algoritmo de contagem quântica.

Dado dois registradores x (de tamanho t q-bits) e y (de tamanho necessário para conter um estado quântico qualquer $|u\rangle$), a aplicação da estimativa de fase permite obter o ângulo de fase φ relativo a $|u\rangle$ com erro ξ. O dimensionamento de x está relacionado ao erro de forma que, sendo n a precisão desejada em q-bits para a estimativa, x deve possuir o tamanho t dado por

$$t = n + \left\lceil \log_2 \left(2 + 1/2\xi\right) \right\rceil .$$

Por exemplo, para estimar a fase φ de um estado quântico $|y\rangle$ de três q-bits, pressupõe-se o uso de um registrador $|x\rangle$ com tamanho $n = 3$. Caso se queira estimar a fase com precisão ξ=0.1, o registrador $|x\rangle$ deve ter seu tamanho aumentado para $t = 5.81 \sim 6$. O circuito da Figura 18 representa, portanto, o algoritmo de estimativa de fase neste caso. Após a aplicação da transformação de Walsh-Hadamard ao registrador $|x\rangle$, cada uma das portas faz uma estimativa, mudando o estado de $|x\rangle$ (o estado de $|y\rangle$ permanece inalterado). Por último, aplica-se a transformada inversa de Fourier para obter a estimativa de fase φ presente no estado $|y\rangle$.

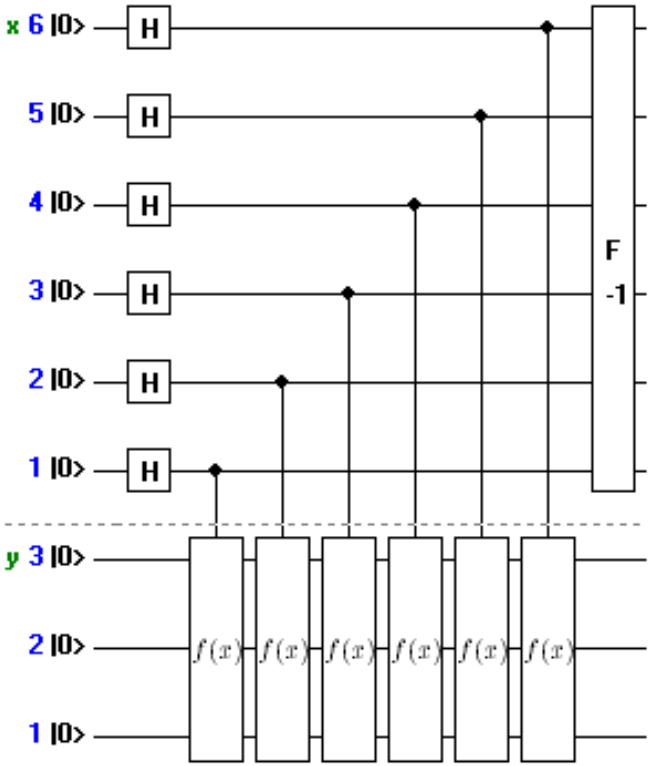


Figura 18: Exemplo de circuito para estimativa de fase.
Fonte: Elaborado pelo autor.

2.5.3 Contagem Quântica

A contagem quântica é um algoritmo que combina os algoritmos de Grover e a estimativa de fase, permitindo **a obtenção do número de soluções** num determinado problema de busca, com certa margem de erro (NIELSEN e CHUANG, 2005; BRASSARD et al, 1998). O algoritmo de Grover, tal como explanado na seção 2.5.1, permite a busca de um estado quântico dentre um conjunto de estados com complexidade $O(\sqrt{N})$. Entretanto, o algoritmo também pode ser

utilizado para encontrar mais de uma solução de forma probabilística, com a complexidade do algoritmo ficando em $O(\sqrt{N/M})$, onde *M* é o número de soluções. **A contagem quântica pode fornecer, antes de se buscar uma solução específica com o algoritmo de Grover, uma estimativa de quantas soluções *M* existem para um dado problema de busca**.

As operações na busca de Grover consistem na aplicação do oráculo mais um deslocamento de fase para evidenciar o estado quântico procurado. Estas duas operações combinadas na realidade aplicam uma rotação sobre o vetor de busca no espaço onde a base é constituída pelo vetor alvo mais o conjunto dos vetores restantes. O vetor alvo é, portanto, rotacionado por um ângulo (fase) θ, onde (NIELSEN e CHUANG, 2005)

$$\sin\left(\theta/2\right) = \sqrt{M/2N}$$

Com base nesta relação e o tamanho *N* conhecido, pode-ser estimar a fase θ para se encontrar a quantidade de soluções *M*. O circuito quântico deve ser dimensionado de acordo com a equação (51) vista na estimativa de fase.

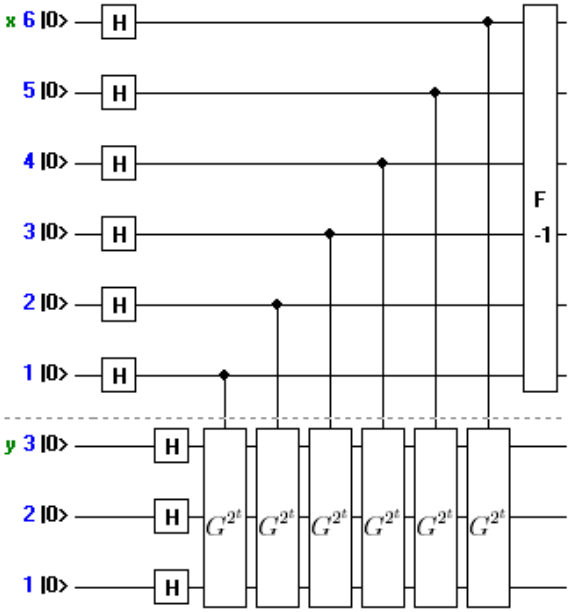


Figura 19: Circuito para contagem quântica do exemplo.
Fonte: Elaborado pelo autor.

Por exemplo, para estimar a quantidade de soluções para um problema com N=8 contendo M=3 soluções (desconhecidas) de um estado quântico preparado $|y\rangle$ de três q-bits, supõe-se o uso de um registrador $|x\rangle$ com tamanho n = 3. Para estimar a fase com precisão ξ=0.2, o registrador $|x\rangle$ deve ter seu tamanho aumentado para t = 5.17 ~ 6. Assim, o circuito da Figura 19 mostra, portanto, o algoritmo de contagem quântica combinando a estimativa de fase com a aplicação do algortimo de Grover na faixa de valores de acordo com o tamanho de $|x\rangle$. Após a aplicação da transformação de Walsh-Hadamard ao registrador $|x\rangle$, cada uma das portas faz uma estimativa, mudando o estado de $|x\rangle$ (ao estado de $|y\rangle$ é aplicada uma iteração do algoritmo de Grover). Por último, aplica-se a transformada inversa de Fourier para obter a estimativa do ângulo θ presente no estado $|y\rangle$.

Em caso de não existirem soluções, o ângulo θ assume valor zero, indicando que M=0. O algoritmo de contagem quântica também pode

ser utilizado, desta forma, para **indicar a existência ou não** de soluções em um dado problema de busca (NIELSEN e CHUANG, 2005).

2.5.4 Simuladores de Circuitos Quânticos

O desenvolvimento de simuladores para computação quântica auxilia no entendimento e na construção de circuitos quânticos. Tais simuladores utilizam o formalismo quântico para executar as operações sobre q-bits de acordo com a teoria. Outros simuladores existentes são o QCE (DE RAEDT et al, 2000), que trabalha com diferentes paradigmas de implementação de circuitos quânticos; Zeno (CABRAL et al, 2005; BARBOSA et al, 2006), que apresenta os estados de acordo com o formalismo da notação de Dirac; e jQuantum (DE VRIES, 2006), que implementa as operações quânticas na forma de circuitos em registradores com tamanho máximo de 15 q-bits.

2.5.5 Memória Quântica

Para a inicialização de algoritmos ou ainda o armazenamento de valores sendo processados, é necessário que exista uma arquitetura provida de memórias para tal armazenamento. Nielsen e Chuang (2005), estendendo o conceito da busca quântica em bancos de dados, colocam dois tipos de armazenamento em memória que podem ser implementados em circuitos quânticos: i) uso de *memória quântica* que contenha $N = 2^n$ células com *l* q-bits cada um, armazenando os itens de banco de dados $|d_x\rangle$; e ii) uso de *memória clássica* contendo $N = 2^n$ células com *l* q-bits cada um, armazenando os itens de banco de dados d_x. A diferença principal do mecanismo de uma memória quântica residiria em uma característica peculiar de endereçamento, através de um índice $|a\rangle$ em um estado de superposição e permitindo, dessa forma, que **uma superposição de estados quânticos seja carregada da memória** (NIELSEN e CHUANG, 2005). Para implementar este algoritmo de busca em memória, são necessários 3 registradores mais um q-bit (aquele utilizado como auxiliar na busca de Grover) e assim

$$|a\rangle\,|s\rangle\,|0\rangle\,|-\rangle\,,$$

onde o estado do q-bit auxiliar é $|-\rangle = \big(|0\rangle - |1\rangle\big)/\sqrt{2}$. A operação de carregar da memória coloca, então, o circuito quântico no seguinte estado,

$$|a\rangle|s\rangle|d_a\rangle|-\rangle .$$

A seguir, entra a operação do oráculo, e o segundo e o terceiro registrador são comparados. Caso sejam iguais, a fase do circuito é modificada, senão fica como está. Assim, serão duas situações:

$$|a\rangle|s\rangle|d_a\rangle|-\rangle \rightarrow \begin{cases} -|a\rangle|s\rangle|d_a\rangle|-\rangle, \text{ se } d_a = s \\ |a\rangle|s\rangle|d_a\rangle|-\rangle \text{ , se } d_a \neq s \end{cases}.$$

Após o oráculo, o deslocamento de fase de Grover é aplicado para permitir a obtenção do dado buscado.

Nielsen e Chuang comentam que, para que o oráculo funcione corretamente com superposições, parece que a memória precisa ter características quânticas. Tal proposta pode funcionar com uma memória clássica também, desde que exista uma espécie de **endereçamento quântico**. O esquema de endereçamento proposto por eles necessitaria de $O(N \log N)$ chaves quânticas para endereçar as $N = 2^n$ células (NIELSEN e CHUANG, 2005).

Giovannetti et al (2008a; 2008b) apresentaram uma arquitetura de memória quântica denominado de *QRAM – Quantum Random Access Memory*. A *QRAM* permitiria o acesso aleatório das $N = 2^n$ células de memória distintas, com redução de forma exponencial utilizando $O(\log N)$ chaves quânticas ao invés das *N* requeridas em designs tradicionais de memórias (clássicas ou quânticas). Obtém-se então uma memória quântica mais robusta, requerendo emaranhamento com menos portas quânticas e uma redução exponencial do poder de endereçamento. A *QRAM* pode fazer o acesso em memória endereçando uma superposição coerente dos dados.

2.5.6 Linguagens de Programação Quânticas

Para se lidar com o hardware quântico de forma similar ao paradigma convencional de software, têm-se realizado o desenvolvimento de linguagens e ferramentas computacionais para simular e testar resultados previstos pela teoria com poucos q-bits. Knill (1996) propôs inicialmente uma das primeiras linguagens de alto nível trabalhando com pseudocódigo. Esta linguagem estava baseada em uma arquitetura denominada por ele de QRAM (Máquina Quântica de Acesso Aleatório). Esta arquitetura diferia um pouco da arquitetura

proposta por Deutsch (1985), que trabalhou essencialmente com um modelo de máquina de Turing quântica. A arquitetura de Knill consiste numa extensão de uma máquina clássica que faria acesso a uma memória quântica. Este computador clássico faria o pré-processamento e o pós-processamento dos dados necessários para rodar os algoritmos quânticos, e ainda controlar a memória quântica (KNILL, 1996).

Seguindo a linha de Knill, uma proposta que foi desenvolvida por Ömer (1998) é a da linguagem de programação QCL (*Quantum Computation Language*). A ideia principal no desenvolvimento de uma linguagem deste tipo é converter os conceitos e formalismos que embasam a teoria quântica em um paradigma de melhor compreensão pela comunidade da Ciência da Computação. Somando-se a este fato, enumera-se também a falta de técnicas adequadas de programação para lidar com variáveis quânticas locais, gerenciamento de altas quantidades de espaço e comprimentos dinâmicos de registradores. Assim, a QCL tenta preencher a lacuna consistindo numa linguagem de alto nível, invariante quanto à arquitetura dos computadores quânticos, e possuindo uma sintaxe derivada de linguagens procedurais tais como C e Pascal. Estas características permitem, portanto, a completa implementação e simulação de algoritmos quânticos em um único formalismo consistente (ÖMER, 1998). Por exemplo, o Quadro 14 mostra o algoritmo de Deutsch, apresentado na seção 2.4.9, representado na linguagem QCL (VIZZOTO e ROCHA COSTA, 2006).

```
qcl> qureg a[2]
qcl> H(a[1])
qcl> H(a[2])
qcl> cond Uf (a[1],a[2])
qcl> H(a[1])
qcl> measure(a[1])
```

Quadro 14: O algoritmo de Deustch escrito em QCL.
Fonte: Adaptado de Vizzoto e Rocha Costa (2006).

Selinger (2004) apresenta a QPL (Quantum Programming Language) que é fundamentada na abordagem de controle clássico com dados quânticos. A linguagem QPL teve uma sintaxe baseada em diagramas de fluxo com construtos para inicialização, atribuição única, exclusão ou descarte de q-bits, estrutura condicional relativa à medida, permutações e um operador para a união. Ela possui ainda uma semântica denotacional baseada em formalismo quântico para estados mistos, que se baseiam em matrizes de densidade e superoperadores.

Existem apenas procedimentos de primeira ordem, que recebem e retornam bits quânticos (VIZZOTO e COSTA, 2006).

Já a linguagem QML (ALTENKIRCH e GRATTAGE, 2005), é uma linguagem funcional e tem sua abordagem baseada tanto em controle quanto em dados quânticos. Ela apresenta primitivas para controle quântico através de um "*if*" que analisa o dado quântico sem aplicar uma medida sobre ele, permitindo desta forma a avaliação de uma superposição quântica (VIZZOTO e COSTA, 2006). No Quadro 15 é mostrado o algoritmo de Deutsch escrito em QML.

```
deutsch : Q2 × Q2 × Q2
deutsch a b =
let (×, y) = if {qfalse | qtrue }
then (qtrue, if  a
then ({qfalse | (1) qtrue }, (qtrue, b))
else ({(1) qfalse | qtrue }, (qfalse, b)))
else (qfalse, if  b
then ({(1) qfalse | qtrue }, (a, qtrue))
else ({qfalse | (1) qtrue }, (a, qfalse)))
in had ×
```

Quadro 15: O algoritmo de Deutsch na linguagem QML.
Fonte: Adaptado de Altenkirch e Grattage (2005).

3 *FRAMEWORK* PARA ONTOLOGIAS E COMPUTAÇÃO QUÂNTICA CONSIDERANDO PROCESSAMENTO

A elaboração de um modelo combinando ontologias com o paradigma quântico enseja uma abordagem incremental, indo desde formas mais simples considerando a ontologia construída de acordo com a forma convencional, avançando no sentido da construção de ontologias em modo clássico-quântico. A seguir serão abordadas as tarefas de processamento de ontologias complexas utilizando a Computação Quântica: validação de instâncias, raciocínio transitivo e *merging* de ontologias. Tais tarefas estão enquadradas na utilização da superposição de instâncias e superposição de classes.

O modelo de processamento quântico de ontologias operando sob esta ótica está baseado em três premissas:

a) **Arquitetura híbrida**: a arquitetura deve considerar uma forma de armazenamento em memória clássica de uma ontologia. Quando for requerida a execução de algum algoritmo quântico, os dados deverão ser recuperados da memória clássica e armazenados na memória quântica, seguindo a arquitetura de Knill (1996). A partir da medida do sistema, a informação é direcionada à memória clássica (Figura 20). De forma genérica, outro diagrama com arquitetura híbrida de acordo com Ömer (2003), é mostrado na Figura 21. Na Figura 22 está a representação esquemática para a memória quântica a ser utilizada nos algoritmos. Ainda não existe na literatura de Computação Quântica uma representação padronizada para memórias em circuitos quânticos, por isso será adotada aqui uma forma semelhante à representação utilizada para memórias de acesso aleatório (RAM - *Random Access Memory*), inspirada na eletrônica digital. Para o endereçamento são utilizados os q-bits da parte superior do circuito, e para os dados a serem recuperados, os q-bits inferiores. No exemplo dado pela figura, são utilizados 4 q-bits que podem endereçar 2^4=16 estados quânticos com resolução de 3 q-bits, ou seja, 2^3=8 níveis, sendo possível então endereçar dados como estados quânticos na faixa [$|0\rangle$,$|7\rangle$].

b) **Processamento estocástico**: os algoritmos quânticos, mesmo utilizando a superposição de estados do sistema, fornecem

informações probabilísticas a respeito dos resultados. A não ser para responder a consultas tais como a existência ou não de instâncias específicas (sem necessidade de saber alguma em particular), os algoritmos retornam um estado a partir de um conjunto de estados prováveis. Se a busca de uma solução em específico não é requerida, o algoritmo quântico apresenta a vantagem de menor complexidade de tempo.

c) **Mapeamento clássico-quântico**: assume-se que os componentes (classes, relações ou instâncias) de uma ontologia elaborada de forma convencional possam ser representados por estados quânticos. Assim, o mapeamento entre os elementos da ontologia respectiva com estados quânticos distintos precisa ser considerado.

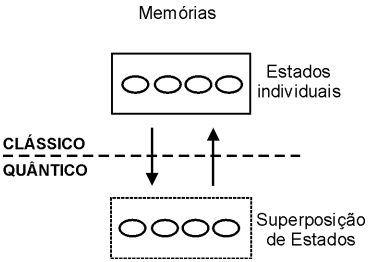


Figura 20: Arquitetura híbrida de memórias para o processamento quântico de ontologias.
Fonte: Elaborado pelo autor.

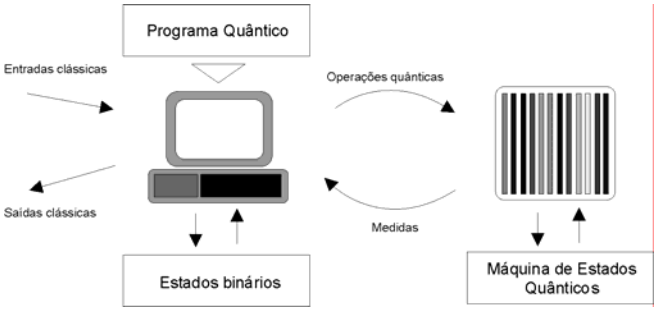


Figura 21: Arquitetura física para processamento.
Fonte: Adaptado de Ömer (2003).

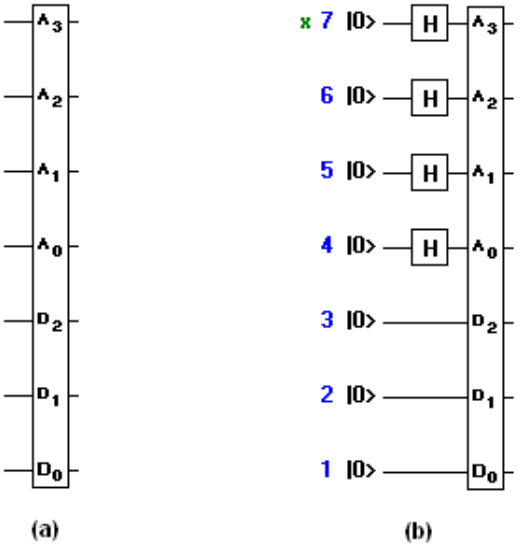


Figura 22: Representação esquemática de uma memória quântica.
Fonte: Elaborado pelo autor.
Nota: Em (a), representação esquemática de uma memória com 4 q-bits de endereçamento e 3 q-bits de dados. Em(b), circuito para acessar uma superposição através do uso de portas Walsh-Hadamard nos q-bits de endereçamento.

3.1 VALIDAÇÃO DE INSTÂNCIAS

Para a tarefa de validação de instâncias sob o processamento quântico, a ontologia é construída de acordo com as metodologias convencionais. Este processamento envolve o uso de elementos da ontologia na forma como estão organizados os conceitos, relações e outros tipos de componentes, sendo transformados a partir da necessidade de alguma tarefa em questão, tal como a validação ou consistência da ontologia, ou a execução de inferências sobre ela. Aqui, o processamento quântico é feito sobre as instâncias existentes para uma dada classe à qual se aplica (ou é permitida) a propriedade da superposição, ou seja, o conceito pode assumir um estado de superposição de suas instâncias.

A validação de ontologias requer que o modelo de conhecimento de um domínio de onde os conceitos foram abstraídos contenha instâncias que estejam consistentes com este modelo (Figura 23). Tal consistência pode ser verificada mediante a existência de axiomas na ontologia que produzam tautologias. Entretanto, instâncias contraditórias podem ser encontradas devido a equívocos ou ainda ao modelo de conhecimento não possuir expressividade suficiente.

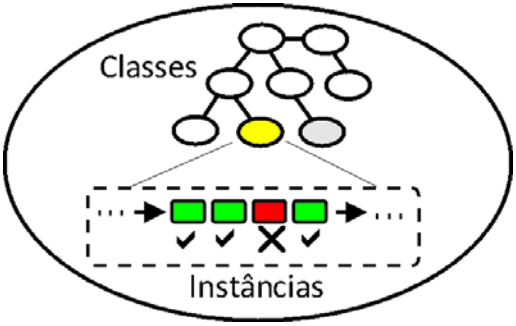


Figura 23: Ilustração da tarefa de validação de instâncias.
Fonte: Elaborado pelo autor.

Para efetuar este tipo de processamento quântico sobre uma ontologia ou parte dela, é necessária uma operação de extração de um grupo de instâncias relativas a um conceito ou conceitos que atendam a um critério específico, que serão mais tarde colocados em superposição no circuito quântico. A representação da ontologia deve permitir a indicação também de qual conceito, daqueles constantes dentro da ontologia, pode ser expresso mediante superposição, indicado assim através de um atributo deste conceito. O algoritmo para validação deve testar os axiomas para todas as instâncias relacionadas aos conceitos envolvidos. O tempo de execução desta tarefa irá depender da quantidade de instâncias relativas aos conceitos e do tamanho e complexidade da ontologia. Um algoritmo quântico pode auxiliar permitindo a recuperação e superposição de todas as instâncias a serem testadas, e a existência ou não de instâncias contraditórias pode ser evidenciada com poucos passos no algoritmo de contagem quântica. Caso existam, tais instâncias podem ser recuperadas de forma estocástica.

Assim, o algoritmo faz uso de axiomas da ontologia que indiquem a satisfação de alguma propriedade ou não das instâncias presentes. Se o conjunto de instâncias satisfaz o axioma, então o

algoritmo retorna zero; caso contrário retorna a quantidade de instâncias que não atendem. Utilizando a expressão da lógica de primeira ordem, para duas variáveis *x* e *y* representando duas classes, a função implementa então uma quantificação existencial da negação do axioma $\exists x, y \mid \neg axioma(x, y)$. Uma função *f*(*x*,*y*) é modelada de forma a retornar o conjunto-verdade onde:

$$\begin{cases} f(x, y) = 1, & \text{caso invalidem o axioma} \\ f(x, y) = 0, & \text{qualquer outro caso.} \end{cases}$$

Esta função será utilizada no algoritmo de contagem quântica na aplicação do oráculo. O circuito utiliza registradores na forma $|i\rangle|j\rangle|x\rangle|y\rangle|a\rangle|e\rangle$, com *x* e *y* sendo os registradores do espaço de trabalho, $|a\rangle$ o q-bit auxiliar para a busca de Grover, e $|e\rangle$ o registrador para o cálculo da estimativa. Dependendo da complexidade do axioma envolvido na validação, a estrutura de registradores bem como seus tamanhos em q-bits podem sofrer alterações. Em Ömer (2003) pode-se encontrar o conceito de **condição quântica** (*quantum condition*) sendo uma fórmula booleana operando sobre um conjunto de q-bits.

O algoritmo utiliza a **contagem quântica** (seção 2.5.3) para estimar o número de soluções *M*. As instâncias com a fase modificada para (-1) serão amplificadas, após a primeira iteração de Grover exigida no algoritmo, aumentando assim a probabilidade de obtenção das instâncias que invalidam o axioma. A estimativa de fase irá permitir o cálculo de *M*, dentro da margem de erro presente no circuito de contagem. No caso de *M*=0, não existem instâncias que invalidam o axioma proposto, sendo o conjunto de instâncias, portanto, válido para a ontologia. Caso contrário, o registrador apontará uma estimativa de quantas instâncias invalidam o axioma. A busca de cada uma pode ser feita posteriormente de forma estocástica, aplicando-se o algoritmo de Grover. Na Figura 24 ilustra-se o funcionamento do algoritmo de validação. O algoritmo "transita" pelos mundos clássico e quântico, com a memória clássica fornecendo as instâncias, acontecendo a seguir a superposição, a consulta ao oráculo e a contagem quântica; finalmente, a medida fornece a informação sobre a consistência ou não da ontologia.

A vantagem deste algoritmo, descrito em detalhe no Quadro 16 em relação a qualquer outro clássico é que, enquanto no algoritmo clássico, a quantificação existencial deve testar as instâncias uma a uma no axioma com complexidade média $O(N)$ (neste algoritmo,

pressupondo um teste condicional simples), o algoritmo quântico exibe complexidade $O(\sqrt{N/M})$.

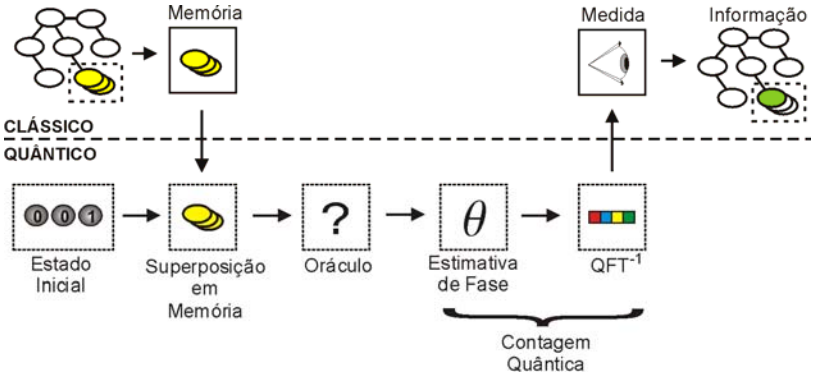


Figura 24: Diagrama esquemático do algoritmo de validação de instâncias.
Fonte: Elaborado pelo autor.

1) **Inicialização do registrador quântico**

$$|i\rangle|j\rangle|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle|a\rangle|e\rangle$$

$|i\rangle$ - estado indexador i
$|j\rangle$ - estado indexador j
$|x_1\rangle|y_1\rangle$ - pares de instâncias associadas indexadas pelo indexador i
$|x_2\rangle|y_2\rangle$ - pares de instâncias associadas indexadas pelo indexador j
$|a\rangle$ - registrador auxiliar para a busca de Grover
$|e\rangle$ - registrador de estimativa utilizado para a contagem quântica

$$|\psi_0\rangle = |0\rangle|0\rangle|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle$$

2) **Transformação de Hadamard do registrador** $|i\rangle$

$$|\psi_1\rangle = H(|0\rangle)|0\rangle|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle = \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N}|i\rangle|x_1\rangle|y_1\rangle\right)|0\rangle|x_2\rangle|y_2\rangle$$

3) **Transformação de Hadamard do registrador** $|j\rangle$

$$|\psi_2\rangle = \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right)H(|0\rangle)|x_2\rangle|y_2\rangle =$$

$$= \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right)\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle|x_2\rangle|y_2\rangle\right)$$

4) **Leitura de memória quântica** para a superposição indexada pelo registrador |i⟩

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right).\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle|x_2\rangle|y_2\rangle\right)$$

5) **Leitura de memória quântica** para a superposição indexada pelo registrador |j⟩

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right).QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle|x_2\rangle|y_2\rangle\right)$$

6) Aplicação do **oráculo**

$$|\psi_5\rangle = O(|\psi_4\rangle) = (-1)^{f(x_1,x_2,y_1,y_2)}|\psi_4\rangle$$

7) Aplicação de um **deslocamento de fase** (de acordo com a busca de Grover)

$$\psi_6 = (2|\psi_5\rangle\langle\psi_5| - I)$$

8) Aplicação da **contagem quântica** (considerando-se o registrador de estimativa |*e*⟩ estando em superposição), *t* vezes

$$|\psi_7\rangle = QFT^{-1}\left[\frac{1}{\sqrt{2^t}}\sum_{e=0}^{2^t-1}|e\rangle U^k(|\psi_6\rangle)\right] = |\hat{e}\rangle|\psi_6\rangle$$

t - tamanho do registrador |*e*⟩ em q-bits (ver seção 2.5.2 Estimativa de Fase)

9) **Medição do registrador de estimativa** |*ê*⟩ *ê*, contendo o número de instâncias (vezes 2, pois os pares de instâncias são colocados dois a dois) que não atendem ao critério de validação do axioma.

Quadro 16: Algoritmo genérico para validação de instâncias.
Fonte: Elaborado pelo autor.

Dependendo da precisão requerida para os estados, os passos 6 e 7 podem ser repetidos, consistindo na iteração de Grover. A contagem quântica dependerá do tamanho *t* do registrador de estimativa. Para se verificar a existência de instâncias que invalidem o axioma, não é necessário um tamanho grande para *t*.

3.2 ALGORITMO PARA RACIOCÍNIO TRANSITIVO

O uso de raciocínio transitivo em uma ontologia permite que se faça a conexão entre classes além da vizinhança próxima em uma ontologia, tornando possível a obtenção de novos conhecimentos, a partir do que está representado na mesma (Figura 25).

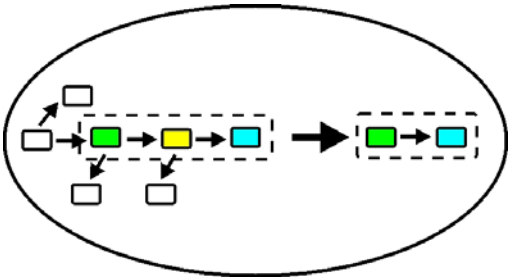


Figura 25: Ilustração da tarefa de raciocínio transitivo.
Fonte: Elaborado pelo autor.

A ideia de utilizar um algoritmo quântico aqui é potencializar esta obtenção, ainda que seja de forma estocástica (Figura 26). O algoritmo clássico precisa trabalhar com a ontologia em memória e o processamento se torna complexo com o aumento do tamanho da ontologia. O uso da propriedade de superposição no algoritmo quântico permitiria colocar em uma memória quântica todas as possíveis inferências transitivas presentes em uma ontologia. A medida do sistema no último passo fornece apenas uma dentre todas as alternativas com certa probabilidade e, portanto, o algoritmo descrito no Quadro 17 deverá ser executado várias vezes para fornecer um conjunto de inferências transitivas.

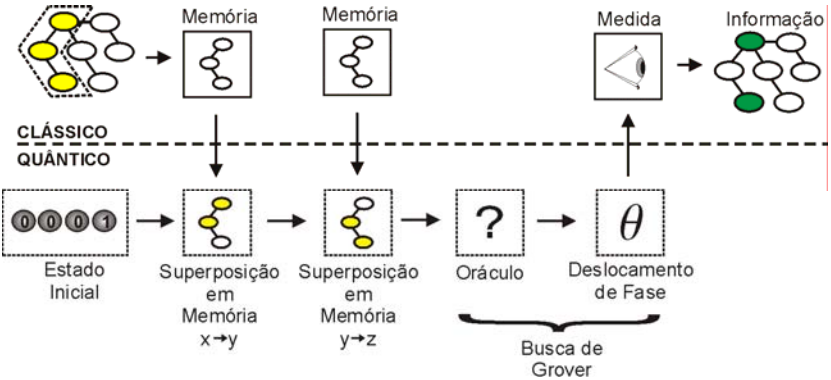


Figura 26: Diagrama esquemático do algoritmo estocástico de raciocínio transitivo.
Fonte: Elaborado pelo autor.

1) **Inicialização do registrador quântico**

$$|i\rangle|j\rangle|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle|a\rangle$$

$|i\rangle$ - estado indexador i
$|j\rangle$ - estado indexador j
$|x_1\rangle|y_1\rangle$ - pares de instâncias associadas indexadas pelo indexador i
$|x_2\rangle|y_2\rangle$ - pares de instâncias associadas indexadas pelo indexador j
$|a\rangle$ - registrador auxiliar para a busca de Grover

$$|\psi_0\rangle = |0\rangle|0\rangle|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle$$

2) **Transformação de Hadamard do registrador** |i⟩

$$|\psi_1\rangle = H(|0\rangle)|0\rangle|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle = \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N}|i\rangle|x_1\rangle|y_1\rangle\right)|0\rangle|x_2\rangle|y_2\rangle$$

3) **Transformação de Hadamard do registrador** |j⟩

$$|\psi_2\rangle = \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right)H(|0\rangle)|x_2\rangle|y_2\rangle =$$

$$= \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right)\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle|x_2\rangle|y_2\rangle\right)$$

4) **Leitura de memória quântica** para a superposição indexada pelo registrador |i⟩

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right).\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle|x_2\rangle|y_2\rangle\right)$$

5) **Leitura de memória quântica** para a superposição indexada pelo registrador |j⟩

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle|x_1\rangle|y_1\rangle\right).QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle|x_2\rangle|y_2\rangle\right)$$

6) Aplicação do **oráculo**

$$|\psi_5\rangle = O(|\psi_4\rangle) = (-1)^{f(x_1,x_2,y_1,y_2)}|\psi_4\rangle$$

$$\begin{cases} f(x_1,y_1,x_2,y_2)=1, & \text{para } (x_1 \,{}^{\backprime}\, x_2)\wedge(y_1 \,{}^{\backprime}\, y_2)\wedge(x_2 = y_1) \\ f(x_1,y_1,x_2,y_2)=0, & \text{qualquer outro caso.} \end{cases}$$

7) Aplicação de um **deslocamento de fase** (de acordo com a busca de Grover)

$$\psi_6 = (2|\psi_5\rangle\langle\psi_5| - I)$$

8) **Medição dos registradores**, retornando de forma estocástica as relações transitivas encontradas.

Quadro 17: Algoritmo genérico para raciocínio transitivo.
Fonte: Elaborado pelo autor.

A complexidade do algoritmo envolve, além das duas buscas dos pares em memória quântica, $O(P\sqrt{N/M})$, com N sendo o número de pares no conjunto considerado, M o número de soluções e P o número de vezes que o algoritmo será rodado para obtenção das relações transitivas de forma estocástica.

3.3 *MERGING* DE ONTOLOGIAS

Outra forma de explorar a superposição de classes é utilizá-la em *merging* (ou apenas o alinhamento) de ontologias (MEDEIROS et al, 2010). Supondo um conjunto único indexado de classes, a partir do qual duas ou mais ontologias fazem uso, é possível a concepção de um algoritmo quântico que explore a superposição das classes, de forma a mostrar a existência de classes comuns às ontologias envolvidas e de maneira estocástica obter tais classes (Figura 27).

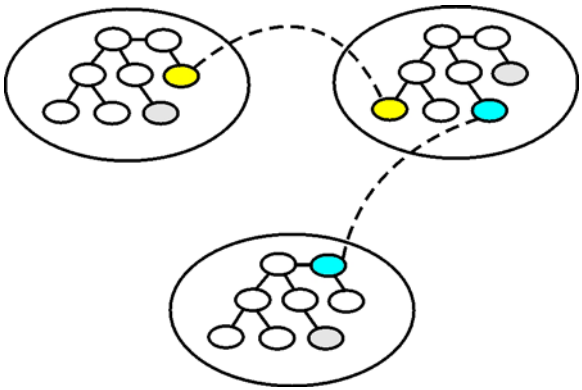


Figura 27: Ilustração da tarefa de *merging* de ontologias.
Fonte: Elaborado pelo autor.

Um melhoramento neste processo pode ser obtido combinando-se os algoritmos de contagem quântica e o algoritmo de busca de Grover.

Na Figura 28 está representado o algoritmo para verificação da existência de classes comuns a duas ontologias, utilizando a contagem quântica. Na Figura 29 é representado o algoritmo para obter de forma estocástica as classes que são comuns às ontologias, agora utilizando a busca de Grover.

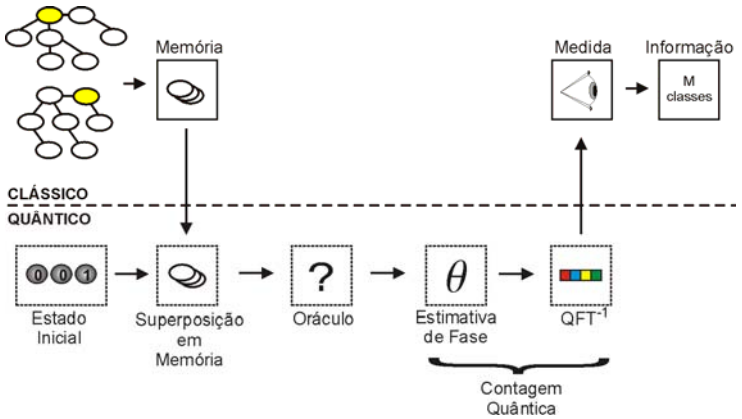


Figura 28: Algoritmo de verificação da existência de classes comuns para duas ontologias, utilizando contagem quântica.
Fonte: Elaborado pelo autor.

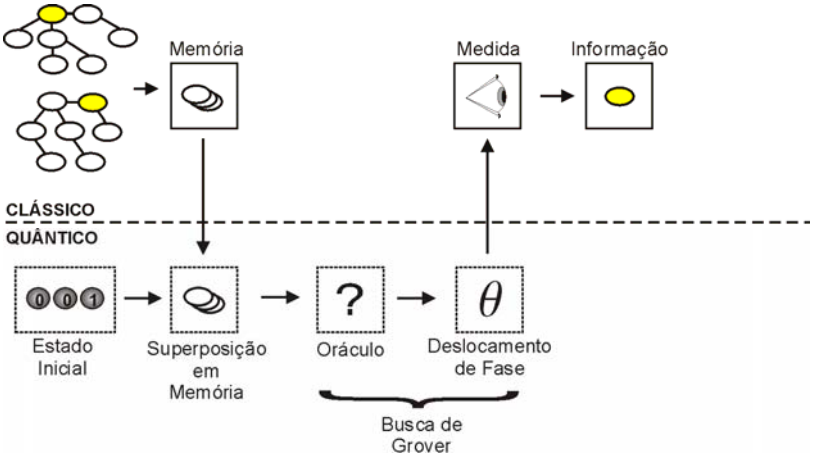


Figura 29: Algoritmo para busca estocástica de classes comuns para duas ontologias, utilizando busca de Grover.
Fonte: Elaborado pelo autor.

No Quadro 18 está descrito o algoritmo para obtenção das classes comuns, de acordo com o diagrama apresentado na Figura 29. Para o diagrama da Figura 28, o algoritmo é bastante semelhante ao apresentado na validação de instâncias.

1) **Inicialização do registrador quântico**

$$|i\rangle|j\rangle|x\rangle|y\rangle|a\rangle$$

$|i\rangle$ - estado indexador i
$|j\rangle$ - estado indexador j
$|x\rangle$ - classes indexadas pelo indexador i
$|y\rangle$ - classes indexadas pelo indexador j
$|a\rangle$ - registrador auxiliar para a busca de Grover

$$|\psi_0\rangle = |0\rangle\,|0\rangle\,|x\rangle\,|y\rangle$$

2) **Transformação de Hadamard do registrador** |i⟩

$$|\psi_1\rangle = H(|0\rangle)\,|0\rangle\,|x\rangle\,|y\rangle = \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\,|x\rangle\right)|j\rangle\,|y\rangle$$

3) **Transformação de Hadamard do registrador** |j⟩

$$|\psi_2\rangle = \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\,|x\rangle\right)H(|0\rangle)\,|y\rangle =$$

$$= \left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\,|x\rangle\right)\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle\,|y\rangle\right)$$

4) **Leitura de memória quântica** para a superposição indexada pelo registrador |i⟩

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\,|x\rangle\right).\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle\,|y\rangle\right)$$

5) **Leitura de memória quântica** para a superposição indexada pelo registrador |j⟩

$$|\psi_4\rangle = QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{i=0}^{N-1}|i\rangle\,|x\rangle\right).QRAM_R\left(\frac{1}{\sqrt{N}}\sum_{j=0}^{N-1}|j\rangle\,|y\rangle\right)$$

6) Aplicação do **oráculo**

$$|\psi_5\rangle = O(|\psi_4\rangle) = (-1)^{f(x_1,x_2,y_1,y_2)}\,|\psi_4\rangle$$

$$\begin{cases} f(x,y) = 1, & \text{para (x = y)} \\ f(x,y) = 0, & \text{qualquer outro caso.} \end{cases}$$

7) Aplicação de um **deslocamento de fase** (de acordo com a busca de Grover)

$$\psi_6 = (2|\psi_5\rangle\langle\psi_5| - I)$$

8) **Medição dos registradores**, retornando de forma estocástica as classes comuns encontradas.

Quadro 18: Algoritmo genérico para merging de ontologias.
Fonte: Elaborado pelo autor.

A complexidade do algoritmo envolve, além das duas buscas dos pares em memória quântica, $O(P\sqrt{N/M})$, com N sendo o número de classes no conjunto considerado, M o número de classes comuns encontradas e P o número de vezes que o algoritmo será rodado para obtenção das classes comuns uma a uma de forma estocástica.

3.4 FRAMEWORK PARCIAL CONTEMPLANDO PROCESSAMENTO

Os três algoritmos apresentados anteriormente podem ser resumidos no framework apresentada na Figura 30, no qual interessa os modos de integração "Ontologias-Computação Quântica" que possam atender ao aspecto de processamento de ontologias. Os algoritmos de validação e raciocínio transitivo se referem a tarefas envolvendo instâncias das ontologias, enquanto que o merging está relacionado a classes.

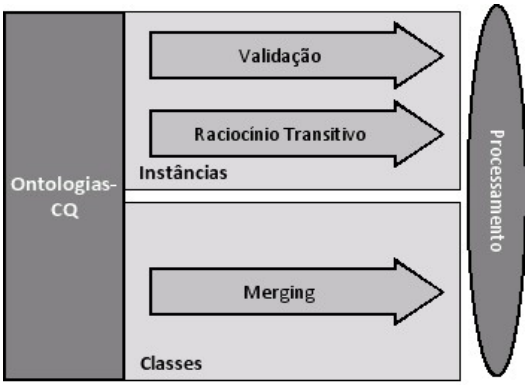


Figura 30: Framework parcial, direcionada às tarefas de processamento de ontologias.
Fonte: Elaborado pelo autor.

Cada um dos algoritmos é explicado a seguir utilizando-se casos práticos.

3.5 CASOS PRÁTICOS DE APLICAÇÃO DOS ALGORITMOS

Um **roteiro** genérico para a explicação a seguir dos três algoritmos de Computação Quântica para as tarefas de validação de instâncias, raciocínio transitivo e *merging* contém:

1) O **caso prático** de ontologia que será resolvido pelo algoritmo, mostrando-se a representação da ontologia e seus elementos;
2) Um quadro com o **mapeamento** dos elementos da ontologia (classes ou instâncias) com estados quânticos a serem utilizados nos algoritmos;
3) Explanação do **algoritmo** em detalhe, com a expressão formalizada do estado quântico na notação de Dirac em cada fase;
4) Apresentação da cláusula lógica para o **oráculo** utilizado no algoritmo, no formato de cláusulas de Horn;
5) Apresentação de um **circuito quântico** sugerido para executar o algoritmo, desenhado no simulador de Computação Quântica;
6) Comparação da **complexidade** do algoritmo quântico em relação à sua contrapartida clássica.

3.5.1 Caso Prático: Validação de Instâncias

O uso do algoritmo de validação é ilustrado com o exemplo de uma ontologia de domínio simples de atendimento (Figura 31). São representados três elementos de ontologia: classes (elipses), instâncias (retângulos) e as relações (conexões entre classes). A ontologia possui então três classes (`C_vendedor`, `C_cliente` e `C_região`) e seis relações (`R_atende`, `R_atendido_por`, `R_pertence_a`, `R_possui`, `R_visita` e `R_visitado_por`). Adotando-se como regra de negócio, o conhecimento será representado de forma consistente nesta ontologia desde que uma instância de `C_vendedor` atue apenas sobre uma instância de `C_região`, que contém instâncias de `C_cliente`. Uma instância de `C_vendedor` deve possuir um conjunto de instâncias de `C_cliente`. Portanto, uma instância de `C_cliente` somente pode ser atendida por uma instância de `C_vendedor`. A consistência desta ontologia é testada mediante o axioma A_1 (representando a relação `R_atende` em forma de cláusulas de Horn):

$$A\_1: \{\neg\exists c, v_1, v_2 \mid R\_atende(v_1,c) \wedge R\_atende(v_2,c) \wedge (v_1 \neq v_2)\}$$

O axioma retorna verdadeiro desde que não exista um cliente *c*, que seja atendido por vendedores v_1 e v_2 distintos entre si. Portanto, este é um teste de consistência para confirmar se as instâncias das classes `C_vendedor` e `C_cliente` estão em relação de *1:N*.

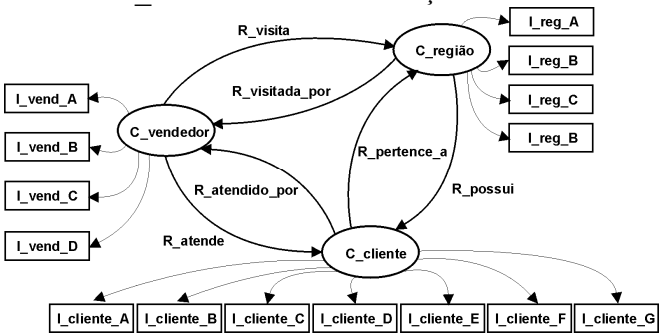


Figura 31: Ontologia de atendimento utilizada no algoritmo de validação.

Fonte: Elaborado pelo autor.

No Quadro 19 são mostradas as instâncias desta ontologia como fatos. A segunda e quarta linhas mostram a inconsistência de um cliente

sendo atendido por dois vendedores. Esta inconsistência deverá ser "adivinhada" pelo oráculo, de forma a ser capturada pela contagem quântica.

No.	C_vendedor	C_região	C_cliente
1	I_vend_A	I_reg_A	I_cliente_A
2	**I_vend_A**	I_reg_A	**I_cliente_B**
3	I_vend_B	I_reg_B	I_cliente_C
4	**I_vend_B**	I_reg_B	**I_cliente_B**
5	I_vend_C	I_reg_C	I_cliente_D
6	I_vend_D	I_reg_D	I_cliente_E
7	I_vend_D	I_reg_D	I_cliente_F
8	I_vend_D	I_reg_D	I_cliente_G

Quadro 19: Instâncias da ontologia de atendimento.
Fonte: Elaborado pelo autor.

A representação em cláusulas de Horn para o oráculo traz dificuldades para a avaliação da função *f*. De forma a contornar o problema, expande-se o espaço de trabalho para que o teste do axioma A_1 seja feito na expressão equivalente A_1a:

$$\texttt{A\_1a}: \{\neg\exists c_1, c_2, v_1, v_2 \mid (c_1 = c_2) \wedge (v_1 \neq v_2)\}$$

Assim, o circuito quântico conterá 7 registradores mais um q-bit auxiliar: $|i\rangle|j\rangle|c_1\rangle|v_1\rangle|c_2\rangle|v_2\rangle|a\rangle|e\rangle$. Os registradores indexadores *x* e *y* terão tamanho N=3. Para fins de simplificação, o q-bit auxiliar $|a\rangle$ não será mostrado. Também estará subentendido o registrador para a estimativa $|e\rangle$, utilizado mais tarde na contagem quântica. Segue-se então a execução do algoritmo:

1) **Inicialização dos registradores** *i* e *j*:

$$|\psi_0\rangle = |0\rangle|0\rangle|c_1\rangle|v_1\rangle|c_2\rangle|v_2\rangle$$

2) **Transformação de Walsh-Hadamard** no registrador *i*, colocando o estado inicial em superposição de oito estados:

$$|\psi_1\rangle = H(|0\rangle)|0\rangle|c_1\rangle|v_1\rangle|c_2\rangle|v_2\rangle = \left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle|c_1\rangle|v_1\rangle\right)|0\rangle|c_2\rangle|v_2\rangle$$

3) **Transformação de Walsh-Hadamard** no registrador *j*, colocando-o também em superposição de 8 estados. Para fins de simplificação, cada registrador indexador será colocado sob

parênteses. Cada estado do indexador deve ser unívoco em relação aos pares de registradores $|c\rangle|v\rangle$, portanto estes são absorvidos para dentro do somatório:

$$|\psi_2\rangle = \left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle|c_1\rangle|v_1\rangle\right)H(|0\rangle)|c_2\rangle|v_2\rangle =$$

$$= \left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle|c_1\rangle|v_1\rangle\right)\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7}|j\rangle|c_2\rangle|v_2\rangle\right)$$

4) **Leitura da memória quântica** dos pares de instâncias das classes vendedor-cliente para o registrador *x*. Cada estado do índice *i* recuperará como dado da memória quântica um par de instâncias associado a um estado quântico específico, num total de 8 estados (de acordo com o Quadro 20).

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle|c_1\rangle|v_1\rangle\right).\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7}|j\rangle|c_2\rangle|v_2\rangle\right) =$$

$$= \frac{1}{2\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|0\rangle + |2\rangle|2\rangle|1\rangle + |3\rangle|1\rangle|1\rangle + |4\rangle|3\rangle|2\rangle + |5\rangle|4\rangle|3\rangle + |6\rangle|5\rangle|3\rangle + |7\rangle|6\rangle|3\rangle\right).$$

$$.\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7}|j\rangle|c_2\rangle|v_2\rangle\right)$$

I	C_cliente	c	C_vendedor	V
$\|0\rangle$	I_cliente_A	$\|0\rangle$	I_vend_A	$\|0\rangle$
$\|1\rangle$	**I_cliente_B**	$\|1\rangle$	**I_vend_A**	$\|0\rangle$
$\|2\rangle$	I_cliente_C	$\|2\rangle$	I_vend_B	$\|1\rangle$
$\|3\rangle$	**I_cliente_B**	$\|1\rangle$	**I_vend_B**	$\|1\rangle$
$\|4\rangle$	I_cliente_D	$\|3\rangle$	I_vend_C	$\|2\rangle$
$\|5\rangle$	I_cliente_E	$\|4\rangle$	I_vend_D	$\|3\rangle$
$\|6\rangle$	I_cliente_F	$\|5\rangle$	I_vend_D	$\|3\rangle$
$\|7\rangle$	I_cliente_G	$\|6\rangle$	I_vend_D	$\|3\rangle$

Quadro 20: Associação das instâncias da ontologia aos estados quânticos.
Fonte: Elaborado pelo autor.

5) **Leitura da memória quântica** dos pares de instâncias das classes vendedor-cliente agora para o registrador *j*. Ao final, faz-se o produto de cada conjunto indexado, totalizando então 64 estados quânticos:

$$|\psi_4\rangle = \frac{1}{2\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|0\rangle + |2\rangle|2\rangle|1\rangle + |3\rangle|1\rangle|1\rangle + |4\rangle|3\rangle|2\rangle + |5\rangle|4\rangle|3\rangle + |6\rangle|5\rangle|3\rangle + |7\rangle|6\rangle|3\rangle\right).$$

$$QRAM\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7} j\rangle|c_2\rangle|v_2\rangle\right) =$$

$$= \frac{1}{8}\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|0\rangle + |2\rangle|2\rangle|1\rangle + |3\rangle|1\rangle|1\rangle + ...\right).\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|0\rangle + |2\rangle|2\rangle|1\rangle + |3\rangle|1\rangle|1\rangle + ...\right) =$$

$$= \frac{1}{8}\left(|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle + ... + |1\rangle|3\rangle|1\rangle|0\rangle|1\rangle|1\rangle + ... + |3\rangle|1\rangle|1\rangle|1\rangle|1\rangle|0\rangle + ... + |7\rangle|7\rangle|6\rangle|3\rangle|6\rangle|3\rangle\right)$$

6) **Aplicação do oráculo** sobre $|\psi_4\rangle$ de forma que

$$\begin{cases} f(c_1, v_1, c_2, v_2) = 1, & \text{para } (c_1 = c_2) \wedge (v_1 \neq v_2) \\ f(c_1, v_1, c_2, v_2) = 0, & \text{qualquer outro caso.} \end{cases}$$

No exemplo, duas soluções para este oráculo serão encontradas e terão suas amplitudes invertidas:

$$|\psi_5\rangle = O(|\psi_4\rangle) = (-1)^{f(c_1,c_2,v_1,v_2)}|\psi_4\rangle =$$

$$= \frac{1}{8}\left(|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle + ... - |\mathbf{1}\rangle|\mathbf{3}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle + ...\right.$$

$$\left. ... - |\mathbf{3}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{1}\rangle|\mathbf{0}\rangle + ... + |7\rangle|7\rangle|6\rangle|3\rangle|6\rangle|3\rangle\right)$$

Na Figura 32 está a primeira parte do circuito quântico que implementa o algoritmo. Apenas ao final do circuito, portas de troca são utilizadas para agrupar os q-bits indexadores[13] $|i\rangle|j\rangle$, necessários para a entrada do circuito de contagem quântica.

7) **Aplicação da contagem quântica** sobre $|\psi_5\rangle$, resultando a estimativa no registrador $|e\rangle$. Para a rotina do algoritmo de Grover presente na contagem quântica, utilizam-se os registradores combinados $|i\rangle|j\rangle$ para indexar a busca (cada registrador contém 3 q-bits, totalizando esta união 6 q-bits, resultando em 2^6=64 estados quânticos).

[13] Esta combinação se refere ao produto tensorial entre os registradores, $|i\rangle \otimes |j\rangle$.

8) **Medida do registrador de estimativa** $|e\rangle$. O algoritmo de contagem quântica encontra uma estimativa de aproximadamente 1.9, com precisão de 5%. A Figura 33 mostra o circuito quântico equivalente para a contagem quântica do exemplo. Com este circuito, deve-se adicionar mais um registrador que deverá mostrar, após a medida, a estimativa das soluções. Na contagem quântica, os registradores $|c_1\rangle|v_1\rangle|c_2\rangle|v_2\rangle$ (que serviram para definir as amplitudes na consulta ao oráculo) são descartados, pois não são necessários ao cálculo da estimativa.

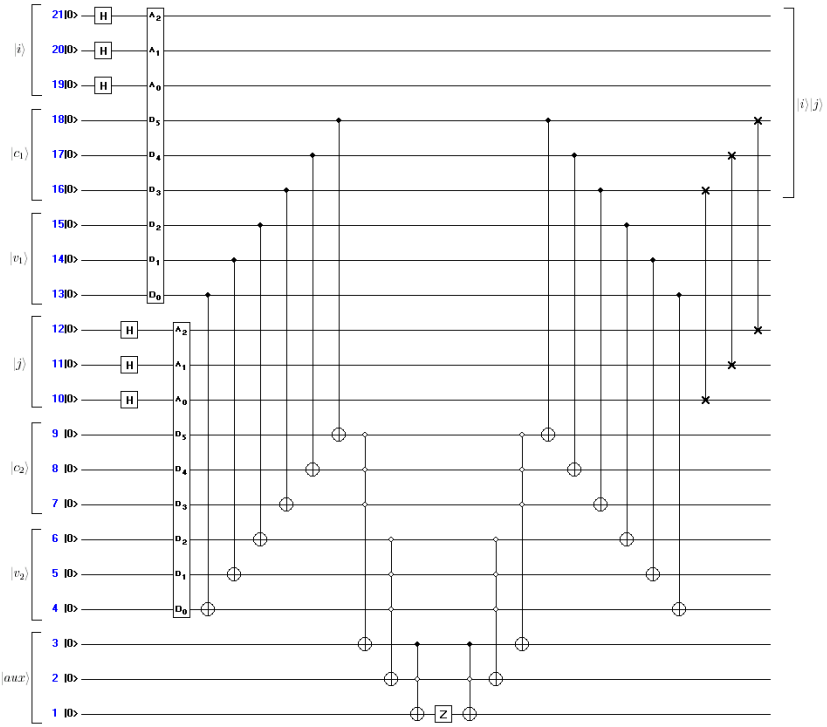


Figura 32: Circuito para a leitura de memória e o oráculo do exemplo da ontologia de atendimento.
Fonte: Elaborado pelo autor.

Nota: O registrador inferior (aux) é utilizado para a operação de comparação e os q-bits de índices 16 a 21 são usados como saída para os dois registradores indexadores $|i\rangle|j\rangle$.

Analisando-se o algoritmo de validação do exemplo, verifica-se que, para a abordagem clássica (Ver Apêndice C), o axioma com o quantificador existencial envolvendo duas cláusulas possui complexidade $O(N^2)$, para varrer todas as instâncias que invalidam a ontologia do exemplo. Exceto a complexidade envolvida na recuperação de dados em memória, o algoritmo quântico continua exibindo $O(\sqrt{N/M})$.

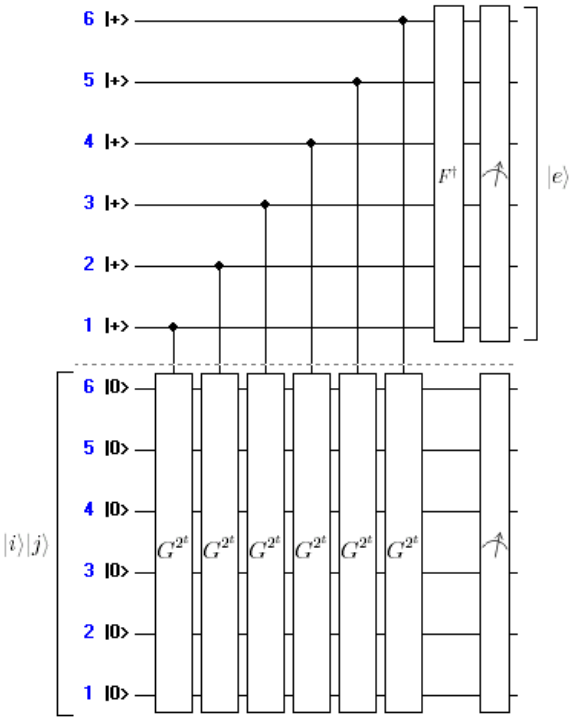


Figura 33: Circuito para contagem quântica do exemplo da ontologia de atendimento.
Fonte: Elaborado pelo autor.
Nota: O registrador inferior (6 q-bits) utiliza como entrada as amplitudes dos dois registradores indexadores $|i\rangle|j\rangle$, provenientes do oráculo, e o registrador $|e\rangle$ terá, após a medida, o valor da estimativa.

3.5.2 Caso Prático: Raciocínio Transitivo

Um caso prático para aplicar o raciocínio transitivo é o uso em diagramas de influência, no qual existem variáveis relacionadas entre si de forma a mostrar impacto da influência de uma sobre outra de forma direta ou inversamente proporcional, tais como os ciclos de realimentação ou balanceamento de Ashby e os modelos utilizados em elaborações de cenários (ANDRADE et al, 2006). Tais modelos refletem a percepção do modelador quanto ao conhecimento de causa e efeito envolvido. Na Figura 34 tem-se uma ontologia de domínio mostrando um exemplo de um diagrama de influência, mostrando a classe `C_variável`, a relação `R_influencia` e as instâncias representando as variáveis `I_var_A` até `I_var_F`, relacionadas entre si. O mapeamento das variáveis para estados é mostrado no Quadro 21.

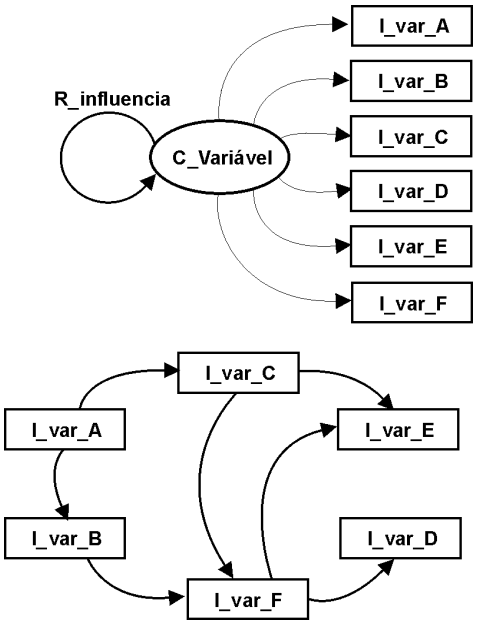


Figura 34: Ontologia para diagrama de influência utilizada como exemplo para o algoritmo de raciocínio transitivo.
Fonte: Elaborado pelo autor.

Causa	Efeito	$\|x\rangle$	$\|r\rangle$
I_var_A	I_var_B	$\|1\rangle$	$\|2\rangle$
I_var_A	I_var_C	$\|1\rangle$	$\|3\rangle$
I_var_B	I_var_F	$\|2\rangle$	$\|6\rangle$
I_var_C	I_var_F	$\|3\rangle$	$\|6\rangle$
I_var_F	I_var_E	$\|6\rangle$	$\|5\rangle$
I_var_C	I_var_E	$\|3\rangle$	$\|5\rangle$
I_var_F	I_var_D	$\|6\rangle$	$\|4\rangle$

Quadro 21: Conversão das relações de influência em estados quânticos
Fonte: Elaborado pelo autor.

Uma análise do problema mostra que se as instâncias fossem associadas de 1 para 1, para cada par de variável causa-efeito, a solução seria simples. Porém, como uma instância de causa pode atuar em duas ou mais instâncias de efeitos, caracteriza-se o problema 1 para muitos. Assim, a abordagem da arquitetura do circuito é semelhante ao exemplo anterior. O circuito conterá 6 registradores, 2 como indexadores, 4 como espaço de trabalho e mais um q-bit auxiliar, na forma $|i\rangle|j\rangle|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle|a\rangle$. Os registradores $|i\rangle|j\rangle$ atuarão como indexadores individualmente na recuperação em memória e combinados para a amplificação de amplitudes. Os registradores $|x_1\rangle|y_1\rangle|x_2\rangle|y_2\rangle$ conterão os dados após duas buscas em memória quântica. Todos os registradores terão tamanho de 3 q-bits. Mais uma vez, o q-bit auxiliar $|a\rangle$ não será mostrado para simplificar a explanação. O índice $|0\rangle$ será associado ao estado $|0\rangle$ na memória quântica. Segue-se então a aplicação do algoritmo ao exemplo de ontologia de diagramas de influência:

1) **Inicialização dos registradores** *i* e *j*:

$$|\psi_0\rangle = |0\rangle\,|0\rangle\,|x_1\rangle\,|y_1\rangle\,|x_2\rangle\,|y_2\rangle$$

2) **Transformação de Walsh-Hadamard** no registrador *i*, colocando o estado inicial em superposição de oito estados:

$$|\psi_1\rangle = H(|0\rangle)\,|0\rangle\,|x_1\rangle\,|y_1\rangle\,|x_2\rangle\,|y_2\rangle = \left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle\,|x_1\rangle\,|y_1\rangle\right)|j\rangle\,|x_2\rangle\,|y_2\rangle$$

3) **Transformação de Walsh-Hadamard** no registrador *j*, colocando-o também em superposição de 8 estados. Para fins

de simplificação, cada registrador indexador será colocado sob parênteses. Cada estado do indexador deve ser unívoco em relação aos pares de registradores $|x\rangle|y\rangle$, portanto estes são absorvidos para dentro do somatório:

$$|\psi_2\rangle = \left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle|x_1\rangle|y_1\rangle\right)H(|0\rangle)|x_2\rangle|y_2\rangle =$$

$$= \left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}|i\rangle|x_1\rangle|y_1\rangle\right)\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7}|j\rangle|x_2\rangle|y_2\rangle\right)$$

4) **Leitura da memória quântica** dos pares de instâncias de causa e efeito para o registrador *i*. Cada estado do índice *i* recuperará como dado da memória quântica um par de instâncias associado a um estado quântico específico, num total de 8 estados (de acordo com o Quadro 21).

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{2\sqrt{2}}\sum_{i=0}^{7}i\rangle|x_1\rangle|y_1\rangle\right).\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7}j\rangle|x_2\rangle|y_2\rangle\right) =$$

$$= \frac{1}{2\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|2\rangle + |2\rangle|1\rangle|3\rangle + |3\rangle|2\rangle|6\rangle + |4\rangle|3\rangle|6\rangle + |5\rangle|6\rangle|5\rangle + |6\rangle|3\rangle|5\rangle + |7\rangle|6\rangle|4\rangle\right).$$

$$.\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7}j\rangle|x_2\rangle|y_2\rangle\right)$$

5) **Leitura da memória quântica** dos pares de instâncias de causa e efeito agora para o registrador *y*. Ao final, faz-se o produto de cada conjunto indexado, totalizando então 64 estados quânticos:

$$|\psi_4\rangle = \frac{1}{2\sqrt{2}}\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|2\rangle + |2\rangle|1\rangle|3\rangle + |3\rangle|2\rangle|6\rangle + |4\rangle|3\rangle|6\rangle + |5\rangle|6\rangle|5\rangle + |6\rangle|3\rangle|5\rangle + |7\rangle|6\rangle|4\rangle\right).$$

$$.QRAM_R\left(\frac{1}{2\sqrt{2}}\sum_{j=0}^{7}j\rangle|x_2\rangle|y_2\rangle\right) =$$

$$= \frac{1}{8}\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|2\rangle + |2\rangle|1\rangle|3\rangle + ...\right).\left(|0\rangle|0\rangle|0\rangle + |1\rangle|1\rangle|2\rangle + |2\rangle|1\rangle|3\rangle + ...\right) =$$

$$= \frac{1}{8}\left(|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle + ... + |1\rangle|3\rangle|1\rangle|2\rangle|2\rangle|6\rangle + ... + |2\rangle|4\rangle|1\rangle|3\rangle|3\rangle|6\rangle + ... + |7\rangle|7\rangle|6\rangle|4\rangle|6\rangle|4\rangle\right)$$

6) **Aplicação de oráculo** sobre $|\psi_4\rangle$ de forma que

$$\begin{cases} f(x_1, y_1, x_2, y_2) = 1, & \text{para } (x_1 \neq x_2) \wedge (y_1 \neq y_2) \wedge (x_2 = y_1) \\ f(x_1, y_1, x_2, y_2) = 0, & \text{qualquer outro caso.} \end{cases}$$

Neste exemplo, sete soluções para este oráculo como relações transitivas serão encontradas e terão suas amplitudes invertidas:

$$|\psi_5\rangle = O(|\psi_4\rangle) = (-1)^{f(x_1,x_2,y_1,y_2)} |\psi_4\rangle =$$

$$= \frac{1}{8}(|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle|0\rangle + ... - |\mathbf{1}\rangle|\mathbf{3}\rangle|\mathbf{1}\rangle|\mathbf{2}\rangle|\mathbf{2}\rangle|\mathbf{6}\rangle + ... - |\mathbf{2}\rangle|\mathbf{4}\rangle|\mathbf{1}\rangle|\mathbf{3}\rangle|\mathbf{3}\rangle|\mathbf{6}\rangle + ...$$

$$... - |\mathbf{2}\rangle|\mathbf{6}\rangle|\mathbf{1}\rangle|\mathbf{3}\rangle|\mathbf{3}\rangle|\mathbf{5}\rangle + ... - |\mathbf{3}\rangle|\mathbf{5}\rangle|\mathbf{2}\rangle|\mathbf{6}\rangle|\mathbf{6}\rangle|\mathbf{5}\rangle + ...$$

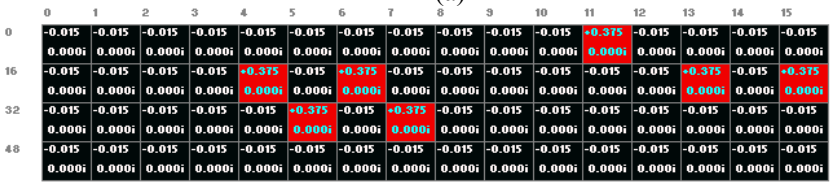
$$... - |\mathbf{3}\rangle|\mathbf{7}\rangle|\mathbf{2}\rangle|\mathbf{6}\rangle|\mathbf{6}\rangle|\mathbf{5}\rangle + ... - |\mathbf{4}\rangle|\mathbf{5}\rangle|\mathbf{3}\rangle|\mathbf{6}\rangle|\mathbf{6}\rangle|\mathbf{5}\rangle + ...$$

$$... - |\mathbf{4}\rangle|\mathbf{7}\rangle|\mathbf{3}\rangle|\mathbf{6}\rangle|\mathbf{6}\rangle|\mathbf{4}\rangle + ... + |7\rangle|7\rangle|6\rangle|4\rangle|6\rangle|4\rangle)$$

7) **Aplicação da rotina** $G = 2|\psi\rangle\langle\psi| - I$ presente na busca de Grover sobre $|\psi_5\rangle$, utilizando-se os registradores combinados[14] $|i\rangle|j\rangle$ para indexar a busca (cada registrador contém 3 q-bits, totalizando a união 6 q-bits, resultando em 2^6=64 estados quânticos). Na Figura 35 está representado o registrador de estados após uma iteração de Grover. A probabilidade de obtenção de uma solução foi amplificada para ~10%, enquanto a de uma não-solução foi atenuada para ~0,5%. Para a segunda iteração, a probabilidade de uma das soluções foi amplificada para ~14%, enquanto a de uma não-solução foi atenuada para ~0,02%. (Aqui, caberia ainda a execução de uma contagem quântica para a identificação do número de soluções e também do número de iterações de Grover necessárias para maximizar a amplitude das soluções buscadas; para M=7 soluções, o algoritmo retornaria R≅2).

[14] Como citado anteriormente, o produto tensorial entre os registradores, $|x\rangle \otimes |y\rangle$.

(a)



(b)

Figura 35: Representação das 7 soluções transitivas dentre os 64 estados, após uma (a) e duas (b) iterações de Grover.
Fonte: Elaborado pelo autor.

8) **Medida dos registradores**, colapsando em uma das soluções (por exemplo, $|3\rangle|5\rangle|2\rangle|6\rangle|6\rangle|5\rangle$). Assim, $x_1 = 2$ e $y_2 = 5$, mostrando que a instância I_var_B causa impacto também na instância I_var_E de forma transitiva.

Este caso demonstrou a forma estocástica de obtenção de relações transitivas na ontologia de influência de causa e efeito. O algoritmo deve ser rodado um dado número de vezes para retornar o conjunto das sete soluções transitivas. A complexidade do algoritmo envolve então, além das duas buscas dos pares em memória quântica, $O(P\sqrt{N/M})$, onde P é o número de vezes que o algoritmo será rodado. Em comparação, o algoritmo clássico de fechamento transitivo otimizado de Warshall (TENENBAUM et al, 1995) no cálculo da matriz[15] de todos os caminhos de comprimento 2, possui complexidade $O(N^2)$, porém N aqui é o número de instâncias envolvidas (Ver Apêndice C). A análise para 6 instâncias resulta em 36 passos no cálculo com o algoritmo de Warshall. Para o algoritmo quântico, N é o número de estados (pelo uso

[15] A informação da relação entre os nós de um grafo precisa estar representada numa *matriz de adjacências*, cujo tamanho é o número de nós neste grafo (TENENBAUM et al, 1995).

combinado $|x\rangle \otimes |y\rangle$, com 64 estados), e com M=7 soluções, a complexidade na busca envolve $\sqrt{64/7} \cong 3$ passos. A quantidade de vezes P que deveria ser rodado o algoritmo quântico, para manter uma vantagem relativa (pois ainda existe o caráter estocástico na obtenção das soluções), seria P=36/3=12 vezes. Portanto, o ganho computacional está fundamentado em uma relação de compromisso entre o número de vezes que o algoritmo quântico será rodado pela probabilidade de obtenção de todas as soluções. O ganho tende a ficar mais evidente com um diagrama de influência mais complexo.

Na Figura 36 é sugerido um circuito quântico para o algoritmo de raciocínio transitivo. A fase do oráculo usa várias portas C-NOT para as compações e um registrador auxiliar de 4 q-bits para a inversão da amplitude dos estados desejados. De forma diferente do algoritmo descrito, no final são mostradas portas de troca para o agrupamento dos q-bits de endereçamento, necessários à aplicação da busca de Grover.

3.5.3 Caso Prático: *Merging* de Ontologias

O mapeamento das variáveis é mostrado no Quadro 22. Na Figura 37 está o exemplo para a união de ontologias, havendo então duas ontologias simples onde uma classe se repete nas duas (C_C). A ideia então é fazer o algoritmo aumentar a probabilidade de obtenção desta classe. O circuito quântico conterá 5 registradores na forma $|i\rangle|j\rangle|x\rangle|y\rangle|a\rangle$, com dois registradores indexadores $|i\rangle|j\rangle$ de tamanho N=2 q-bits, proporcionando 4 estados cada um (compondo ao final 8 estados). Os registradores $|x\rangle|y\rangle$ são o espaço de trabalho para recuperação das classes convertidas em estados na memória quântica. Novamente, o registrador $|a\rangle$ não será mostrado por questões de simplicidade.

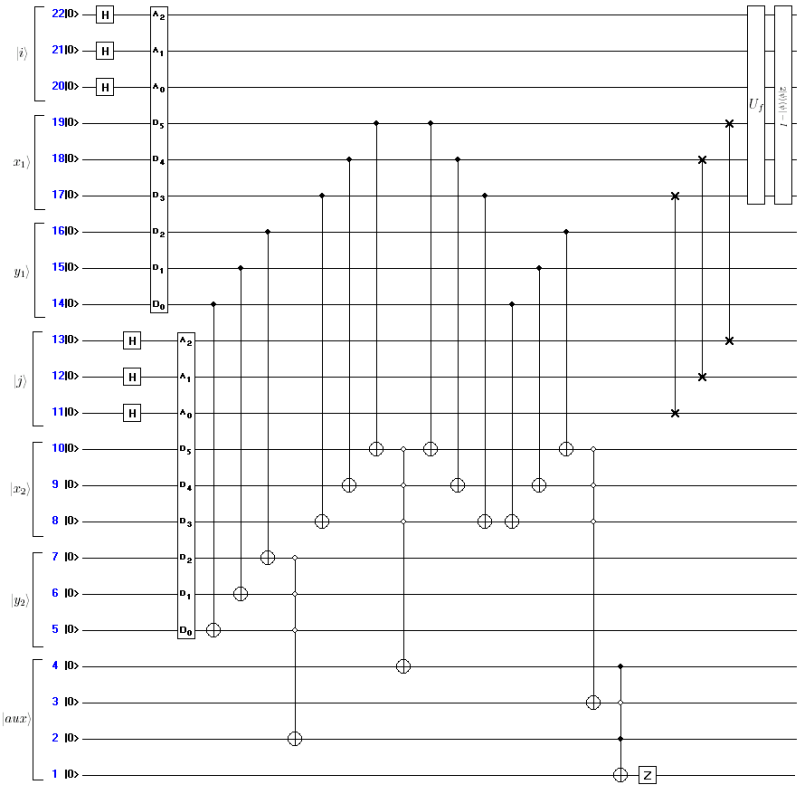


Figura 36: Circuito para a leitura de memória e o oráculo do exemplo da ontologia de raciocínio transitivo.
Fonte: Elaborado pelo autor.
Nota: O registrador inferior (|aux⟩) é utilizado para a operação de comparação e os q-bits de índices 17 a 22 são usados como saída para a busca de Grover.

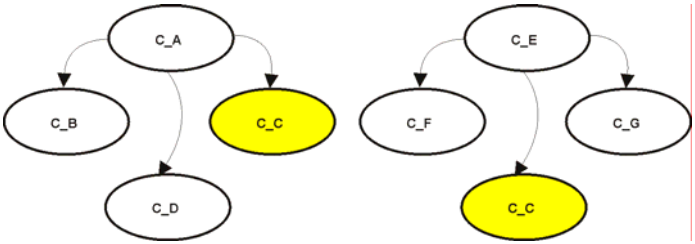


Figura 37: Exemplo para união de ontologias.
Fonte: Elaborado pelo autor.

Classe	Estado
C_A	$\|0\rangle$
C_B	$\|1\rangle$
C_C	$\|2\rangle$
C_D	$\|3\rangle$
C_E	$\|4\rangle$
C_F	$\|5\rangle$
C_G	$\|6\rangle$

Quadro 22: Conversão das classes em estados quânticos.
Fonte: Elaborado pelo autor.

1) **Inicialização dos registradores** *i* e *j*:

$$|\psi_0\rangle = |0\rangle\,|0\rangle\,|x\rangle\,|y\rangle$$

2) **Transformação de Walsh-Hadamard** no registrador *i*, colocando o estado inicial em superposição de quatro estados:

$$|\psi_1\rangle = H(|0\rangle)\,|0\rangle\,|x\rangle\,|y\rangle = \left(\frac{1}{2}\sum_{i=0}^{3}|i\rangle\,|x\rangle\right)|j\rangle\,|y\rangle$$

3) **Transformação de Walsh-Hadamard** no registrador *j*, colocando-o também em superposição de 4 estados. Para fins de simplificação, cada registrador indexador será colocado sob

parênteses. Cada estado do indexador deve ser unívoco em relação ao par de registradores $|x\rangle|y\rangle$, portanto estes são absorvidos para dentro do somatório:

$$|\psi_2\rangle = \left(\frac{1}{2}\sum_{i=0}^{3}|i\rangle\,|x\rangle\right) H(|0\rangle)\,|y\rangle = \left(\frac{1}{2}\sum_{i=0}^{3}|i\rangle\,|x\rangle\right)\left(\frac{1}{2}\sum_{j=0}^{3}|j\rangle\,|y\rangle\right)$$

4) **Leitura da memória quântica** dos pares de instâncias de causa e efeito para o registrador *i*. Cada estado do índice *i* recuperará como dado da memória quântica um par de instâncias associado a um estado quântico específico, num total de 4 estados (de acordo com o Quadro 22).

$$|\psi_3\rangle = QRAM_R\left(\frac{1}{2}\sum_{i=0}^{3}|i\rangle\,|x\rangle\right).\left(\frac{1}{2}\sum_{j=0}^{3}|j\rangle\,|y\rangle\right) =$$

$$= \frac{1}{2}\left(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle + |2\rangle\,|2\rangle + |3\rangle\,|3\rangle\right).\left(\frac{1}{2}\sum_{j=0}^{3}|j\rangle\,|y\rangle\right)$$

5) **Leitura da memória quântica** dos pares de instâncias de causa e efeito agora para o registrador *y*. Ao final, faz-se o produto de cada conjunto indexado, totalizando então 16 estados quânticos:

$$|\psi_4\rangle = \frac{1}{2}\left(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle + |2\rangle\,|2\rangle + |3\rangle\,|3\rangle\right).QRAM_R\left(\frac{1}{2}\sum_{j=0}^{3}|j\rangle\,|y\rangle\right) =$$

$$= \frac{1}{4}\left(|0\rangle\,|0\rangle + |1\rangle\,|1\rangle + |2\rangle\,|2\rangle + |3\rangle\,|3\rangle\right).\left(|0\rangle\,|4\rangle + |1\rangle\,|5\rangle + |2\rangle\,|2\rangle + |3\rangle\,|6\rangle\right) =$$

$$= \frac{1}{4}\left(|0\rangle\,|0\rangle\,|0\rangle\,|4\rangle + \ldots + |2\rangle\,|2\rangle\,|2\rangle\,|2\rangle + \ldots + |3\rangle\,|1\rangle\,|3\rangle\,|5\rangle + \ldots + |3\rangle\,|3\rangle\,|3\rangle\,|6\rangle\right)$$

6) **Aplicação de oráculo** sobre $|\psi_4\rangle$ de forma que

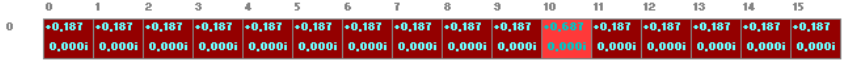
$$\begin{cases} f(x,y) = 1, & \text{para (x = y)} \\ f(x,y) = 0, & \text{qualquer outro caso.} \end{cases}$$

Assim, uma solução para este oráculo contendo a classe comum às ontologias terá sua amplitude invertida:

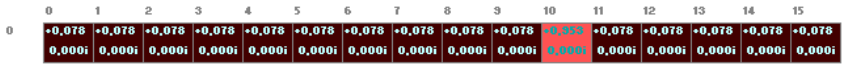
$$|\psi_5\rangle = O(|\psi_4\rangle) = (-1)^{f(x,y)} |\psi_4\rangle =$$

$$= \frac{1}{4}\left(|0\rangle|0\rangle|0\rangle|4\rangle + ... - |\mathbf{2}\rangle|\mathbf{2}\rangle|\mathbf{2}\rangle|\mathbf{2}\rangle + ... + |3\rangle|1\rangle|3\rangle|5\rangle + ... + |3\rangle|3\rangle|3\rangle|6\rangle\right)$$

7) **Aplicação da rotina** $G = 2|\psi\rangle\langle\psi| - I$ presente na busca de Grover sobre $|\psi_5\rangle$, utilizando-se os registradores combinados[16] $|i\rangle|j\rangle$ para indexar a busca (os dois registradores totalizam 4 q-bits, resultando em 2^4=16 estados quânticos). Na Figura 38 está representado o registrador de estados após uma iteração de Grover. A probabilidade de obtenção de uma solução foi amplificada para ~47%, enquanto a de uma não-solução foi atenuada para ~3,4%. Para a segunda iteração, a probabilidade de uma das soluções foi amplificada para ~90%, enquanto a de uma não-solução foi atenuada para ~0,6%. Mais uma vez, poderia ser executada a contagem quântica para a identificação do número de soluções (classes comuns) e também do número de iterações de Grover necessárias para maximizar a amplitude das soluções buscadas; para M=1, o algoritmo retornaria $R \cong 3$.



(a)



(b)

Figura 38: Representação da solução como classe comum dentre os 16 estados, após uma (a) e duas (b) iterações de Grover.
Fonte: Elaborado pelo autor.

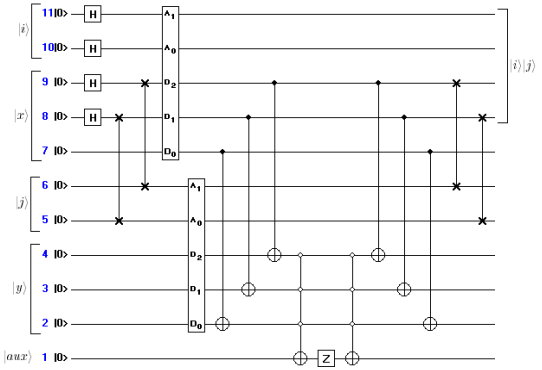
8) **Medida dos registradores**, colapsando na solução $|2\rangle|2\rangle|2\rangle|2\rangle$. Assim, x = 2 e y = 2, mostrando que a classe

[16] Como citado anteriormente, o produto tensorial entre os registradores, $|i\rangle \otimes |j\rangle$.

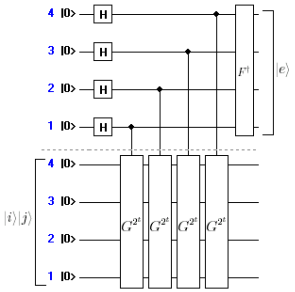
C_C é a classe comum, que pode proporcionar a união das ontologias.

Na Figura 39a é sugerido um circuito quântico de 11 q-bits para o algoritmo de *merging* explicado. Diferente do algoritmo descrito em detalhe, no final do circuito estão colocadas portas de troca para agrupar os q-bits de endereçamento. Este agrupamento é necessário para que a contagem quântica, cujo circuito está mostrado na Figura 39b seja feita. Este circuito de 11 q-bits foi desenhado mas não simulado em virtude da quantidade de q-bits, acima do limite do simulador de Computação Quântica utilizado. Porém, no apêndice B é mostrado um circuito em escala menor, que pôde ser simulado, mostrando o funcionamento correto deste algoritmo.

Este caso demonstrou a forma estocástica de obtenção de classes comuns a duas ontologias. O algoritmo proporcionou apenas uma, porém pode retornar mais de uma classe, caso ela exista. Assim, deve ser rodado um dado número de vezes para retornar o conjunto de tais classes comuns. A complexidade do algoritmo envolve (além da busca dos pares em memória quântica), $O(P\sqrt{N/M})$, onde P é o número de vezes que o algoritmo será rodado. Um algoritmo clássico demandaria complexidade $O(N^2)$, pois cada elemento da primeira ontologia deveria ser testado com a segunda. A análise para uma classe comum resulta em 16 passos no cálculo do algoritmo clássico. Para o algoritmo quântico com N=16 estados e M=1, a complexidade na busca envolve $\sqrt{16/1}=4$ passos. Deve-se considerar aqui (da mesma forma que o exemplo do raciocínio transitivo) a quantidade de vezes P que deveria ser rodado o algoritmo quântico, devido ao caráter estocástico na obtenção das soluções. Mesmo utilizando um algoritmo clássico de busca binária (e assim, a lista deveria antes ser indexada; no caso do algoritmo quântico, não é necessário), a complexidade seria $O(N\log_2 N)$, resultando em 8 passos. O ganho relativo continuaria sendo do algoritmo quântico.



(a)



(b)

Figura 39: Circuito quântico para o algoritmo de *merging*.
Fonte: Elaborado pelo autor.
Nota: Em (a), o oráculo com as portas de troca; em (b), a contagem quântica.

3.6 SIMULADOR PARA COMPUTAÇÃO QUÂNTICA

Na maior parte da explanação foram utilizados, além do formalismo matemático para a explicação, exemplos dos circuitos quânticos aos quais os algoritmos se referem. Para auxiliar no desenvolvimento dos algoritmos apresentados nesta tese, foi criado um **software para simulações de circuitos quânticos**, sendo a interface apresentada na Figura 40. Este software permitiu a modelagem dos algoritmos e simulações de processamento de circuitos com até 11 q-

bits. Na parte superior está o menu de opções de portas quânticas para colocação nos circuitos. No centro existe a área de trabalho, onde é feito o design dos circuitos. À esquerda, estão as opções de execução dos circuitos sendo montados. O software conta ainda com janelas para visualização dos estados quânticos em sua evolução, conforme a execução passo-a-passo do circuito. Na Figura 41 está a representação do estado quântico de três q-bits $|000\rangle$. Lembrando que este estado, na verdade, é a forma compacta da representação do produto tensorial dos três q-bits, $|0\rangle \otimes |0\rangle \otimes |0\rangle$, que na forma equivalente vetorial:

$$|000\rangle = |0\rangle \otimes |0\rangle \otimes |0\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} \otimes \begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

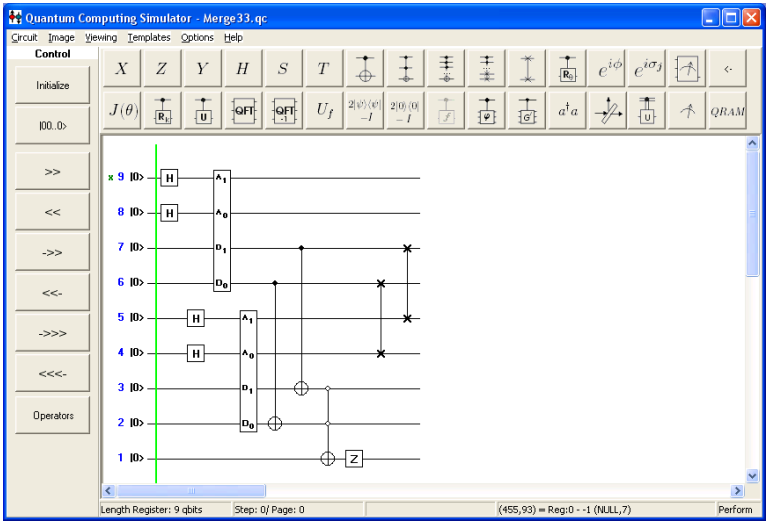


Figura 40: Simulador de Computação Quântica desenvolvido para representação dos algoritmos quânticos.
Fonte: Elaborado pelo autor.

Cada estado quântico possui uma parte **inteira**, indicada na parte superior do quadrado indicando o estado quântico; e a parte **complexa** na parte inferior. A cor vermelha para a representação do valor complexo do estado |0⟩ está de conformidade com a representação de números complexos utilizando o padrão de cores CIE 1931 (HOFFMAN, 2000), com o comprimento do vetor calculado a partir do módulo do número complexo, e o ângulo referente ao eixo *x*-*y* obtido através do argumento do valor complexo. O software também apresenta uma janela de visualização dos estados quânticos, coincidente com este padrão de cores. O mesmo estado referente à expressão 13 é mostrado na Figura 42 nesta forma de representação, com o estado indicado pelo vetor em preto.

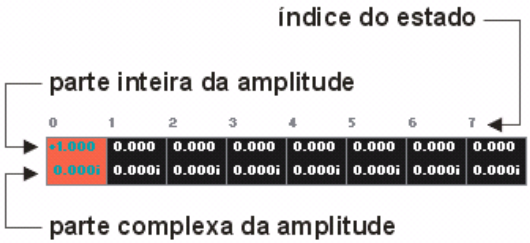


Figura 41: Representação para três q-bits dos oito estados (2^3), equivalente à representação vetorial conforme a equação 13.
Fonte: Elaborado pelo autor.

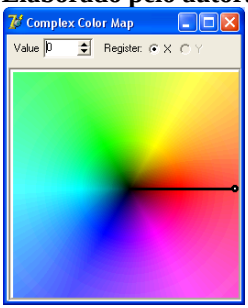


Figura 42: Mapa complexo de cores no padrão HSL coincidente com a representação de números complexos no plano *x*-*y* para o estado |000⟩.
Fonte: Elaborado pelo autor.

O software simulador de Computação Quântica permite também a visualização de um estado quântico na esfera de Bloch (Figura 43).

Um recurso bastante útil para a elaboração dos algoritmos foi o visualizador de estados quânticos na notação de Dirac, cuja tela pode ser visualizada na Figura 44.

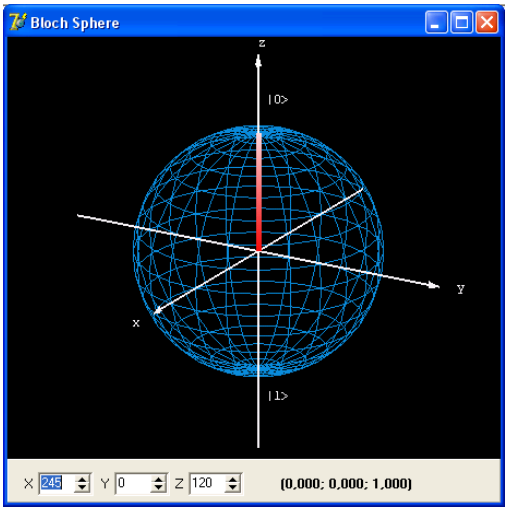


Figura 43: Representação na esfera de Bloch do vetor |0⟩.
Fonte: Elaborado pelo autor.

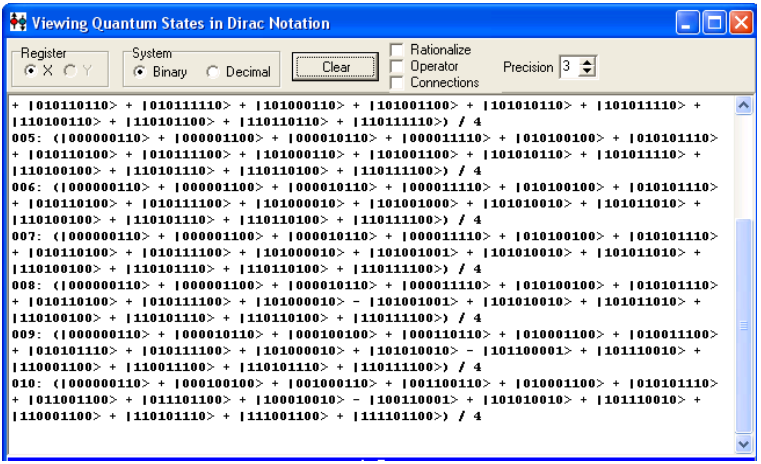


Figura 44: Tela para visualização dos estados quânticos de acordo com a notação de Dirac.
Fonte: Elaborado pelo autor.

4 EVOLUÇÃO DO *FRAMEWORK* PARA ASPECTOS DE ENGENHARIA

Após a proposta de algoritmos quânticos para as tarefas de processamento sobre ontologias construídas de forma convencional, descreve-se agora a própria construção de ontologias acontecendo em um regime clássico-quântico. O regime clássico-quântico para os algoritmos exigia um grau de **armazenamento temporário quântico** de elementos de ontologias. Avançando um pouco mais, explora-se essa assertiva na necessidade de um **armazenamento "mais" permanente**. A ideia por trás disto, além de facilitar o próprio processamento de algoritmos, é dar espaço a outras possibilidades que podem emergir em função de uma representação mais elaborada de ontologias. O *framework* de processamento apresentado anteriormente evolui, e novas formas de engenharia de ontologias são abordadas através de derivações como a superposição de classes e relações e ainda o emaranhamento de classes.

4.1 REPRESENTAÇÃO DE CLASSES SUPERPOSTAS

Neste ponto é retomada a discussão relativa ao consenso em ontologias (seção 2.3.3). Como descrito anteriormente, o conceito de consenso possui uma forte analogia com o problema do modelo conceitual-formal, considerando-se o problema do desenvolvimento da ontologia como um problema de decisão sobre quais classes ou relações, referentes a um modelo conceitual, serão formalizados. O poder de representatividade precisa ser aumentado a partir de tipos específicos de representação ontológica que possam ter um desempenho melhor no limiar dos modelos conceitual-formal, antes de ter lugar o processo de consenso.

Reforçando esta analogia, a formalização e o processo de consenso também estão muito próximos do processo de colapso da função de onda relativa à Mecânica Quântica. Quando o processo de medição é feito sobre uma superposição de estados quânticos, haverá o colapso para apenas um dos estados com certa probabilidade (NIELSEN e CHUANG, 2004). Caso se queira a minimização da aleatoriedade, pode-se utilizar o algoritmo de amplificação de amplitudes quânticas para aumentar as chances de obtenção de algum estado em particular. Portanto, a formalização, o consenso e a medição são processos

similares e fortemente relacionados neste contexto (Figura 45). A última parte da analogia endossa assim o desenvolvimento e o uso de um tipo específico de representação de ontologia, baseado no princípio da superposição, atuando no limiar entre modelos conceituais e formais de ontologias.

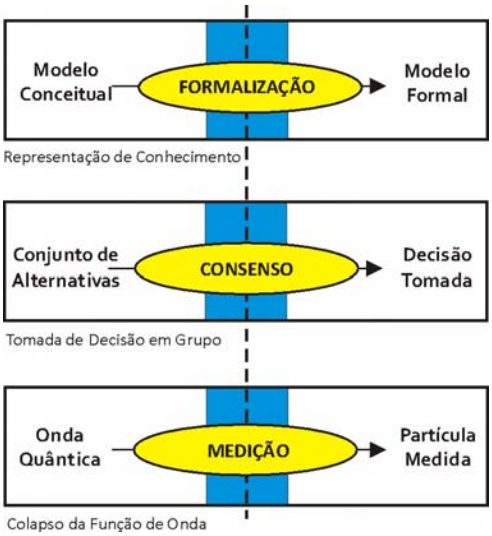


Figura 45: A analogia entre o processo de formalização, o consenso e a medição quântica.
Fonte: Elaborado pelo autor.

Este recurso pode ser uma solução para situações de conflito ou inconsistências presentes na construção da ontologia. Quando na construção de uma ontologia, o engenheiro da ontologia pode enfrentar situações de incerteza, nas quais não dispõe da informação necessária para caracterizar bem uma classe ou um grupo de classes. Nestes casos, adotar uma representação de superposição de classes é conveniente. Até que venha a emergir a informação, pela qual se caracterizará bem uma classe ou outra, as classes estarão em superposição, apresentando probabilidades iguais de ocorrência. É interessante ressaltar que esta espécie de "operador" de superposição é exatamente o que acontece quando se lida com um sistema quântico. Quando não se observa, não se tem a informação. No instante em que se mede o sistema (ou seja,

quando se procura obter a informação sobre alguma característica do objeto, que define este objeto), ele assume apenas uma das prováveis alternativas (Figura 46).

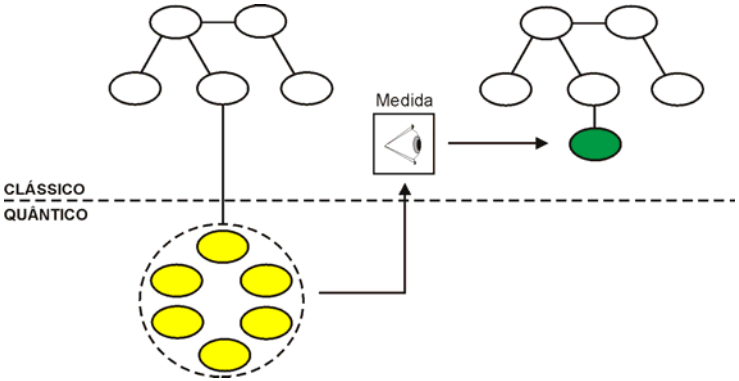


Figura 46: Representação de classes superpostas numa ontologia.
Fonte: Elaborado pelo autor.

Como exemplo, uma parte de uma ontologia de controle acadêmico que pode originar um modelo relacional para construção de um banco de dados é apresentada na Figura 47. A classe C_aluno contém relações bem definidas com as classes `C_professor` e `C_turma`. Porém, relativo às classes `C_curso` e `C_disciplina` a definição não é clara. Duas situações podem ser modeladas aqui:

a) Supondo que a relação seja entre `C_aluno` e `C_curso`, cada aluno pertenceria a um curso que, por sua vez, conteria as disciplinas. Desta forma, estaria implícito a relação 1:N de `C_curso` para `C_disciplina` (Figura 47a).

b) Supondo que a relação seja entre `C_aluno` e `C_disciplina`, cada aluno pertenceria a uma disciplina; com o foco em disciplina, cada curso teria um rol de disciplinas que deveriam ser cursadas, e uma mesma disciplina poderia fazer parte de mais de um curso. Assim, a relação entre `C_curso` e `C_disciplina` seria de cardinalidade *N:N* (Figura 47b).

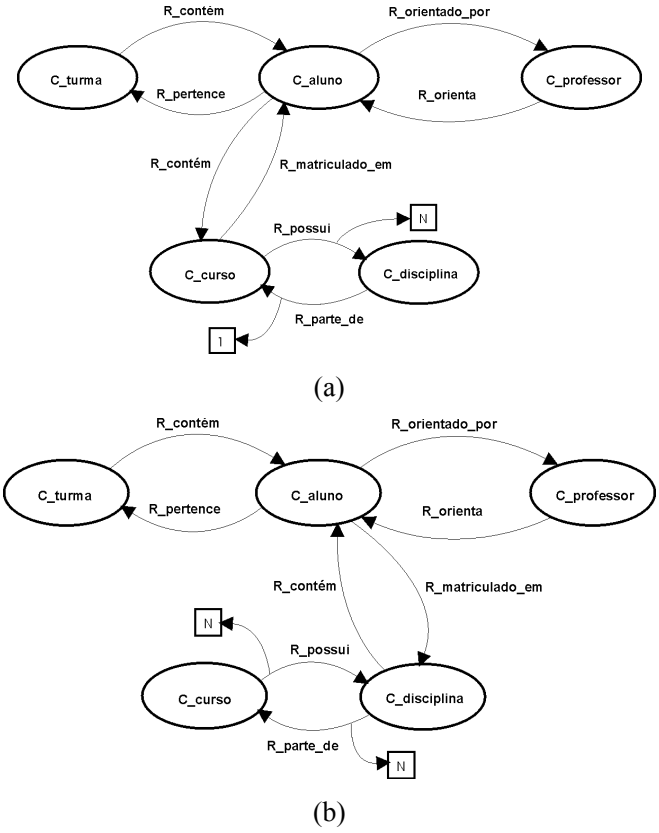


Figura 47: Ontologia-exemplo de controle acadêmico em duas situações.
Fonte: Elaborado pelo autor.

Enquanto não houvesse a escolha de um ou outro modelo de acordo com o consenso da modelagem, faz-se uso de uma *classe superposta* das duas classes, `C_curso` e `C_disciplina` (Figura 48).

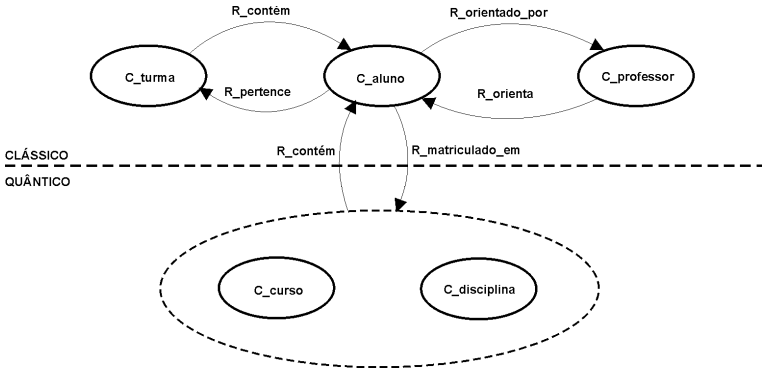


Figura 48: Ontologia-exemplo de controle acadêmico com classes superpostas.
Fonte: Elaborado pelo autor.

A representação de uma classe superposta não ensejaria apenas uma espécie de regra ou filosofia de construção, em alguma metodologia ou ferramenta de elaboração de ontologias. Caracterizaria adicionalmente uma situação com possibilidade de processamento posterior dinâmico do modelo da ontologia, em alguma arquitetura que contemplasse também uma forma híbrida de computação clássica e quântica. A medida da classe superposta resultaria, portanto, em apenas uma alternativa. A escolha pode fazer evoluir o modelo da ontologia, aceitando-se para versões futuras apenas a opção escolhida; ou ainda, se o uso de uma ou outra é trivial, não trazendo impacto significativo para a ontologia modelada, pode-se manter a classe superposta, com o processamento posterior do modelo escolhendo entre uma ou outra classe presente na superposição. A classe superposta seria, portanto, um artifício para acomodar diferentes consensos em uma ontologia, passível de interpretação dinâmica posterior ou inferências em um computador clássico-quântico.

4.2 SUPERPOSIÇÃO DE RELAÇÕES

Outra forma é considerar uma ontologia com um subconjunto de classes no qual elas estão totalmente ligadas entre si, quando não se dispõe da informação de quais relações caracterizam as conexões neste subconjunto da ontologia. Portanto, ao invés dos conceitos, neste caso a superposição de estados se daria com as relações, que teriam probabilidades iguais de ocorrência na ontologia (Figura 49). No instante em que uma relação da ontologia é bem definida, ou seja,

definem-se as classes envolvidas na relação, isto corresponderia à observação na qual o estado quântico colapsa para a sua efetivação como uma relação presente na ontologia.

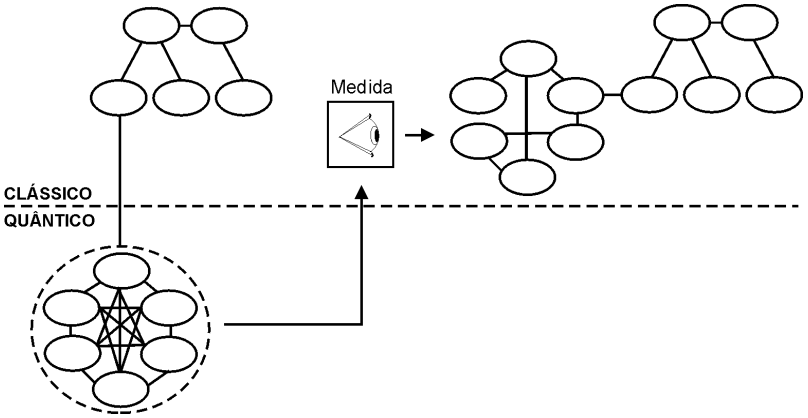
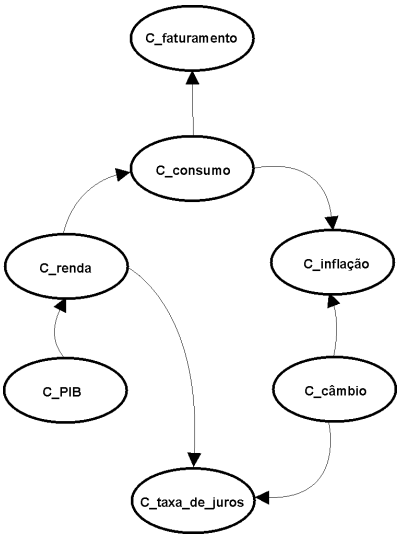
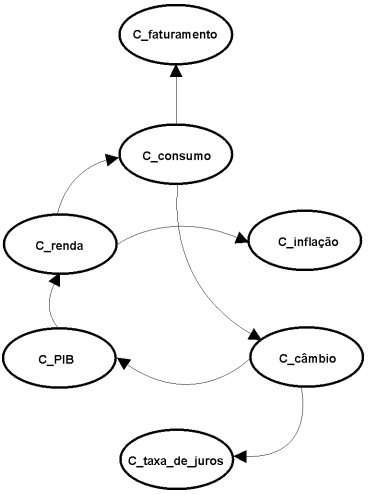


Figura 49: Representação de superposição de relações, resultando em um conjunto específico de relações entre um determinado conjunto de classes.
Fonte: Elaborado pelo autor.

Exemplificando a superposição de relações, a ontologia da Figura 50 se refere a uma parte de uma ontologia utilizada para identificar relações de causa e efeito (semelhante ao exemplo do raciocínio transitivo, porém agora envolvendo classes e não instâncias) com variáveis para construção de lógica de cenários, que impactam no resultado de uma organização. Estão envolvidas sete classes (`C_consumo`, `C_renda`, `C_PIB`, `C_inflação`, `C_taxa_de_juros`, `C_câmbio` e `C_faturamento`) e apenas uma relação, que mostra o impacto direto que uma classe tem em outra, simbolizada pelo sinal positivo. Como tal relacionamento entre as variáveis depende da percepção dos modeladores envolvidos, configurações diferentes de conexões podem surgir (Figura 50a e Figura 50b).



(a)



(b)

Figura 50: Ontologia de exemplo para lógica de cenários refletindo duas percepções diferentes.
Fonte: Elaborado pelo autor.

Uma forma de resolver os conflitos entre várias interpretações é usar **relações superpostas**, onde várias relações estão presentes (ou

todas as possíveis). Até que se atinja o consenso, a ontologia em regime clássico-quântico mantém as relações superpostas. Em complemento à ideia da superposição, pode-se atribuir amplitudes de forma a privilegiar as relações presentes em mais de um modelo. A atribuição das amplitudes (ou algum meio de amplificação-atenuação das mesmas) poderia seguir alguns algoritmos já vistos na Computação Quântica, como a busca de Grover. O processamento posterior das relações trabalharia de forma dinâmica com probabilidades de ocorrência de relações, em um computador clássico-quântico. A Figura 51 mostra a ontologia considerando relações superpostas ligando todas as seis classes referentes às variáveis na ontologia de exemplo.

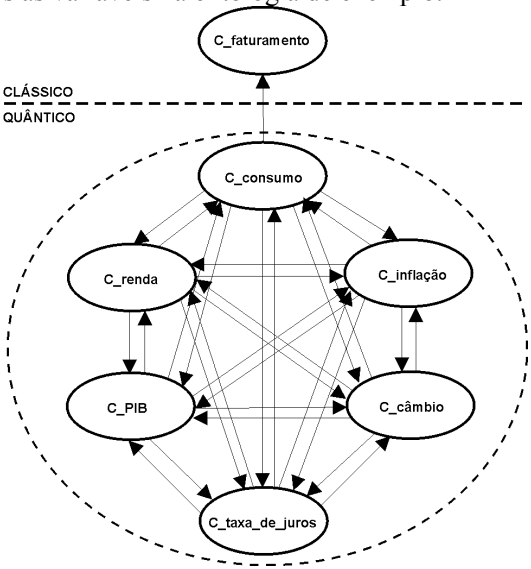


Figura 51: Superposição de relações para a ontologia-exemplo sobre lógica de variáveis.
Fonte: Elaborado pelo autor.

4.3 EMARANHAMENTO DE CLASSES

A disjunção de classes é um recurso bastante explorado nas ferramentas de ontologias, possibilitando a colocação de restrições em classes onde uma é disjunta de outra. A presença de axiomas com disjunção facilita a checagem das instâncias da ontologia. No caso em que duas classes são disjuntas entre si, uma possibilidade é o uso de

estados emaranhados tais como os estados de Bell vistos na seção 2.4.10. Um estado de Bell para representar a disjunção é o estado β_{01}:

$$\beta_{01} = \frac{|01\rangle + |10\rangle}{\sqrt{2}}$$

O estado de emaranhamento garante que na medida de um q-bit, se soubermos o estado dele (0 ou 1), saberemos também, sem necessidade de medida, o estado do outro q-bit emaranhado (Figura 52). Esta propriedade viabiliza uma representação única para dois conceitos disjuntos. Estados emaranhados mais complexos, tais como os estados GHZ que envolvem três q-bits, podem ser utilizados para representar uma quantidade maior de classes disjuntas entre si.

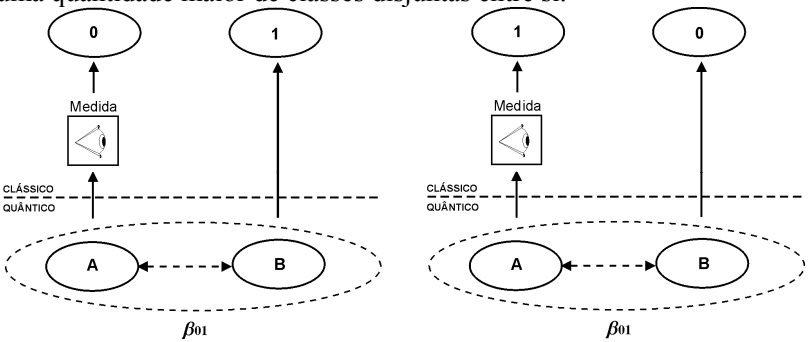


Figura 52: Classes disjuntas representadas pelo estado emaranhado β_{01}.
Fonte: Elaborado pelo autor.

O conceito de classes emaranhadas pode ser estendido também para o caso abordado anteriormente, relativo à **inferência indutiva** ou ainda **lógica de relacionamento de variáveis**. A avaliação do estado de uma variável final a partir do estado de uma variável qualquer relacionada a ela pode ser feito de forma bastante econômica. Supondo uma lógica de variáveis como a da Figura 53 onde existam relações tipificadas de forma positiva e negativa. Para caracterizar a relação positiva, cada variável deve conter o mesmo estado (0 e 0, ou 1 e 1). Por conseqüência, na relação negativa, cada variável contém o estado inverso (0 e 1, ou 1 e 0). Um algoritmo deve fazer a instanciação para verificar o estado de cada variável, chegando-se até a obtenção do estado da variável final. De acordo com a figura, pode-se verificar a similaridade da ontologia e cada par de classes, unidas por relações, como um conjunto de regras de produção:

$$F \Rightarrow G$$
$$F \Rightarrow \neg E$$
$$E \Rightarrow \neg C$$
$$C \Rightarrow \neg D$$
$$C \Rightarrow B$$
$$B \Rightarrow A$$

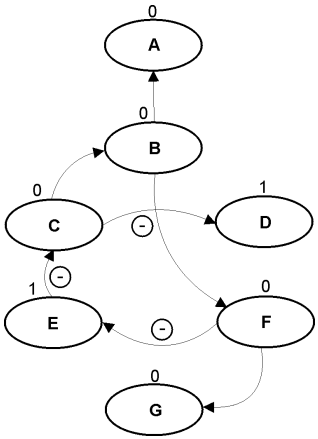


Figura 53: Exemplo de lógica de variáveis para classes emaranhadas. Fonte: Elaborado pelo autor.

Para a realização deste conjunto de regras de produção, começa-se com a premissa $\neg F$, e o processo de indução levará o estado final para $\neg A$. Visando o uso de um algoritmo quântico estocástico, e assumindo que o estado de uma classe ou variável X seja 1 e o estado $\neg X$ seja 0, pode-se associar um conjunto de q-bits emaranhados dispostos na forma $|ABCDEFG\rangle$ para a produção dos estados finais A e $\neg A$. Para a produção de A (estado 1), tem-se:

$$|0001100\rangle$$

Enquanto que para produzir $\neg A$ (estado 0), tem-se a situação com os q-bits invertidos:

$$|1110011\rangle$$

Portanto, ao medir-se o sistema, a chance de obtenção de cada estado deve ser de 50%. O estado total ψ deve ser representado então por:

$$|\psi\rangle = \frac{|0001100\rangle + |1110011\rangle}{\sqrt{2}}$$

A observação da variável do q-bit representado por *F* irá resultar de forma direta no resultado do q-bit representado por *A*. No algoritmo clássico, cada relação significando uma regra deve ser avaliada; havendo *N* relações, a complexidade de avaliação do algoritmo será $O(N)$. No caso do algoritmo quântico usando o emaranhamento, a complexidade se reduz a $O(1)$, ou seja, apenas uma leitura do estado de uma variável (que retornará de forma estocástica 0 ou 1) irá informar o estado final da variável-alvo. Um circuito quântico sugerido para produzir o estado emaranhado para o exemplo da Figura 53 está representado na Figura 54, contendo as portas C-NOT e a porta Walsh-Hadamard (além da porta de medição).

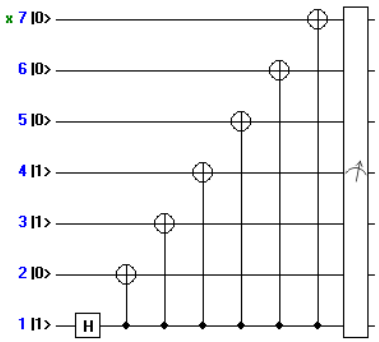


Figura 54: Circuito quântico para a produção da superposição do exemplo.
Fonte: Elaborado pelo autor.

4.4 *FRAMEWORK* GERAL ONTOLOGIAS-CQ

O *framework* da integração entre ontologias e Computação Quântica agora contempla, além do processamento, o aspecto de engenharia de ontologias. Na Figura 55, as possibilidades completam o framework à esquerda com a construção de classes, classes disjuntas e emaranhamento de classes, além da construção de relações. Outra forma de se visualizar o *framework* pode mostrar também as características provenientes de ontologias (classes, instâncias e relações) em conjunto com as propriedades da Computação Quântica (superposição e emaranhamento). Tal framework é diagramado em formas concêntricas e pode ser visualizado na Figura 56. Portanto, têm-se quatro diferentes derivações desta abordagem:

1) **Superposição de instâncias**: explora o princípio da superposição para processar instâncias em tarefas de validação ou raciocínio;
2) **Superposição de classes**: explora o princípio da superposição na construção de ontologias considerando as classes, bem como em tarefas de união ou *merging*;
3) **Superposição de relações**: explora o princípio da superposição na construção de ontologias considerando as relações;
4) **Emaranhamento de classes**: explora o conceito de emaranhamento para caracterizar classes disjuntas ou relacionadas em termos de raciocínio indutivo numa ontologia.

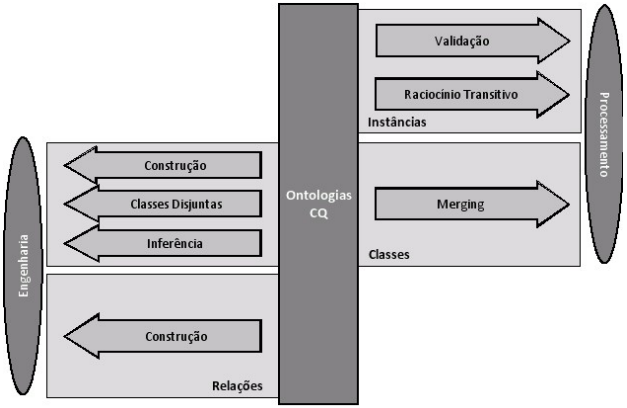


Figura 55: Evolução do *framework* mostrando agora as tarefas considerando a engenharia de ontologias.
Fonte: Elaborado pelo autor.

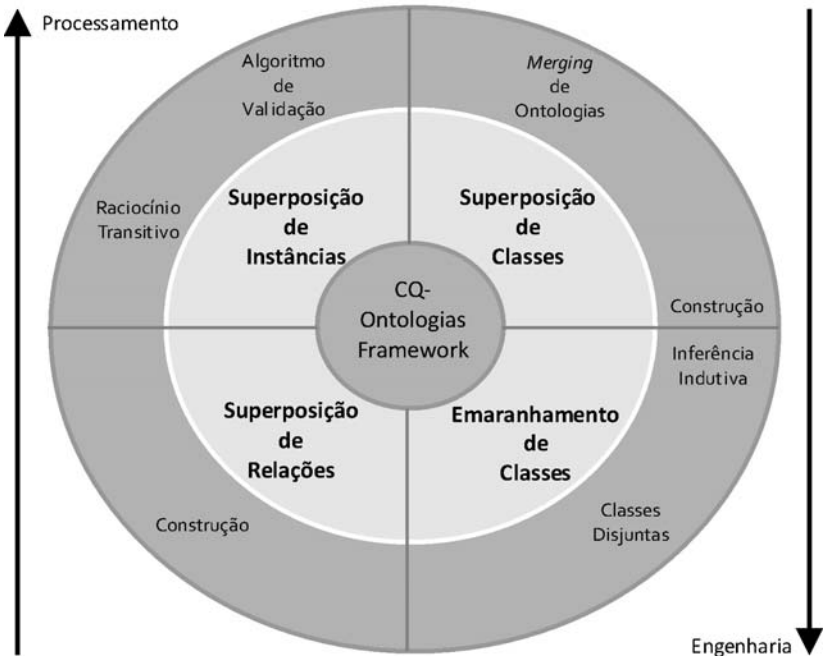


Figura 56: *Framework* para o modelo CQ-Ontologias.
Fonte: Elaborado pelo autor.

A seta no sentido do **processamento** indica um uso prático da abordagem quântica através da Computação Quântica, enquanto que a seta no sentido da **engenharia** busca o aumento do poder de expressividade em ontologias, agregando-se operadores de superposição e emaranhamento, os quais não existem na abordagem formal de ontologias.

4.5 CONSIDERAÇÕES

Neste tópico, abre-se uma discussão sobre vários aspectos gerados a partir das derivações obtidas da união das áreas de Ontologias e Computação Quântica. Tais considerações levantam uma série de questões que não serão abordadas com maior profundidade aqui. Porém, devido à gênese do conceito de ontologias fundamentar-se em aspectos do pensamento simbólico, abre-se o espaço também para a interação

com outras áreas tais como a Inteligência Artificial, podendo se constituir em um elemento indutor para trabalhos posteriores.

A utilização de uma **arquitetura diferenciada** de ontologias que opera tanto no mundo clássico como no quântico levanta uma série de questões. Da utilização dos recursos quânticos emerge um paradoxo com relação à forma clássica de obtenção de informação, que por definição é elaborada e compreendida a partir de dados. Não se observa o que está acontecendo até que se faça a medida do sistema. Os dados devem ficar ocultos, sendo manipulados pelas operações quânticas de forma determinística. Quando acontece a medida de um circuito quântico, o que se converte em informação é apenas uma das possibilidades de obtenção de um resultado, associada com certa probabilidade bem definida. No paradigma clássico, o observador faz emergir a informação a partir de um processo de interpretação e contextualização de forma consciente dos dados. Por conseguinte, no paradigma quântico, parte do processo no qual a informação emerge se dá mediante o processamento de dados de forma **inconsciente**, relativo ao papel do observador. Numa paráfrase à afirmação[17] de Polanyi quanto ao conhecimento tácito, *sempre existe mais informação num sistema quântico do que se pode medir*.

No processamento quântico, o circuito retira vantagem do paralelismo quântico, manipulando todas as instâncias **simultaneamente**. Os circuitos clássicos precisariam, na melhor das hipóteses, trabalhar com redundância de circuitos para chegar ao mesmo resultado, considerando um número pequeno de instâncias, mas com maior gasto de energia e recursos.

Quando em superposição, cada instância tem associada a si uma probabilidade referente a uma opção de resultado do algoritmo. Na interpretação da complementaridade da Mecânica Quântica, a superposição se refere a uma função de onda com todas as instâncias possíveis, relacionada ao registrador quântico. A medida provoca o **colapso** desta função de onda, resultando assim em apenas uma das instâncias como resultado do processamento do algoritmo quântico (a instância se tornaria após a medida, portanto, uma "**partícula**", com ou posição ou momento bem definido, de acordo com o princípio da incerteza de Heisenberg).

No uso da superposição de classes, fica parecendo que a construção de uma ontologia no modo clássico-quântico está mais para uma abordagem de **pensamento quântico** ou uma postura filosófica

[17] "Sempre sabemos mais do que podemos dizer" (POLANYI, 1966).

menos embasada na realidade clássica por parte do construtor da ontologia, do que a aplicação propriamente dita de um processamento quântico nos moldes vistos no modelo da superposição de instâncias. É claro que a ontologia em regime clássico-quântico pode, depois de construída, apresentar conceitos nos quais suas instâncias podem entrar em estado de superposição, e o problema então acaba por recair em ontologias clássicas. Mas se um computador quântico conseguir manter os estados quânticos numa relação de tempo de execução vs. tempo de descoerência alto, o modo quântico de uma ontologia poderia ser armazenado e recuperado de uma memória quântica, levando em consideração os estados superpostos e o emaranhamento, e assim ser processada dinamicamente de acordo com a necessidade da tarefa pretendida. Neste modelo de ontologias, **o limite entre o clássico e o quântico é ampliado**, de forma a incorporar na própria representação de conhecimento da ontologia os aspectos peculiares ao mundo quântico.

A acomodação de diferentes consensos em uma ontologia através de classes ou relações superpostas coloca em questão o próprio conceito de ontologia, no que tange ao argumento da conceituação como resultado de um consenso. A ampliação do escopo da modelagem para o mundo quântico prescinde desta afirmação de consenso, no instante em que a superposição permite uma infinidade de modelos de ontologias ou partes delas presentes ao mesmo tempo. Studer et al (1998) conceituam ontologia como "uma especificação formal e explícita de uma conceituação compartilhada". O compartilhamento reflete a noção de que uma ontologia captura conhecimento consensual, a partir de diferentes percepções de um grupo. Mas o consenso pressupõe uma evolução da ontologia onde, para sua elaboração, conceitos foram confrontados, alguns sendo selecionados e outros descartados. Este descarte proveniente de percepções errôneas poderia ser prejudicial para o alcance de uma ontologia realmente representativa de um domínio. Portanto, na ontologia clássica existe uma **perda natural** no seu processo de construção. Uma ontologia no modo clássico-quântico permitiria o compartilhamento com conhecimento não consensual, e a superposição possibilitaria o acesso simultâneo às várias alternativas.

Algo relevante de menção neste ponto é o fato de se processar algoritmos de raciocínio sobre ontologias em um **regime quântico**. A abstração de ontologias é uma evolução dos *frames* que foram utilizados na área de IA, de acordo com a linha simbólica sugerida nesta área de conhecimento. A linha simbólica nasceu da metáfora da qual a mente humana se utiliza de símbolos e relações entre eles para representar e

adquirir conhecimento (RUSSELL e NORVIG, 2004). Nesta linha de pensamento, pode-se considerar a abordagem aqui descrita como alinhada à proposta dualista de Popper e Eccles (1977), que considera o "mundo" dos eventos mentais (o chamado "mundo 2") como sendo separado do "mundo" dos processos cerebrais ("mundo 1"), ainda que Eccles (1990) subtraia do paradigma quântico apenas a ação probabilística sobre os eventos mentais, baseada no princípio da incerteza (KAK, 1995).

Tal abordagem de ontologias, construídas e processadas de modo quântico, constituiria-se ainda numa analogia paralela à abordagem de Penrose (1991) sobre a existência de fenômenos quânticos nos neurônios como suporte aos processos de raciocínio no cérebro (este seguindo, portanto, uma linha conexionista). Entretanto, a Computação e Informação Quântica consistem numa abstração "digital" de fenômenos quânticos. Caso existam tais fenômenos (afirmado aqui apenas a título de especulação) configurando, portanto, esta situação uma contrapartida simbólica ao modelo de Penrose, a complexidade envolvida nas interações quânticas simbólicas deve acontecer numa escala muito mais complexa do que a proporcionada pela representação mais simplificada da Computação Quântica, a partir de estados quânticos convertidos em q-bits.

5 CONCLUSÃO E RECOMENDAÇÕES

5.1 CONCLUSÃO

Como conclusão da tese, fica visualizada a possibilidade de uso da Computação Quântica como uma ferramenta para o processamento e a engenharia de ontologias. As derivações obtidas da convergência entre as áreas distintas mostram o potencial interdisciplinar, com o *framework* aqui apresentado podendo servir de ponto de partida para aplicações mais complexas, tanto em termos de desenvolvimento de novos algoritmos quanto formas de modelagem inovadoras que possam estar fundamentadas na teoria da Computação Quântica.

O algoritmo quântico de **validação de instâncias** baseado na busca de Grover serviu como um ponto de partida para a análise de ontologias complexas. Operando em um regime clássico-quântico, é necessário que o algoritmo trabalhe com o princípio de superposição para mostrar suas vantagens e o potencial de uso. Este algoritmo, embora utilizando um registrador quântico de alta ordem, permite a redução da complexidade quanto à verificação de inconsistências em ontologias relativas a instâncias fora do contexto das classes. Apesar de o caso prático ter sido demonstrado com o teste de relação de cardinalidade 1:N, pode-se explorar o uso com outros tipos de axiomas. Mostrou-se também a necessidade de memória em modo quântico que pudesse armazenar as instâncias na forma de pares.

O algoritmo quântico para o **raciocínio transitivo** também mostrou o potencial de identificação de relações transitivas entre instâncias, podendo ser estendido para classes de ontologias. Apesar do modo de funcionamento do algoritmo ser estocástico, seu uso em ontologias complexas pode fazer emergir relações transitivas de forma inesperada, caracterizando assim uma espécie de "heurística" baseada em aleatoriedades quânticas na extração de conhecimento a partir de ontologias.

Para a necessidade de ***merging*** ou apenas o alinhamento de ontologias, o algoritmo quântico de *merging* fornece uma alternativa que também lida com a complexidade de ontologias de forma eficiente. Desde que exista uma forma de mapeamento comum de termos relativos às classes nas ontologias envolvidas, a tarefa de junção por meio de classes comuns é viabilizada. Podem-se visualizar as vantagens no uso a partir de grandes ontologias que tenham sido geradas para propósitos diferentes, onde um algoritmo clássico tenha dificuldades de mapear

eficientemente todas as classes que fazem a "fronteira" entre as ontologias.

O desenvolvimento de um **software simulador** de Computação Quântica para os testes dos algoritmos aqui trabalhados foi fundamental, no sentido da elaboração de protótipos de circuitos utilizando um número menor de q-bits. Os algoritmos elaborados aqui trabalham com registradores quânticos com grande número de q-bits. Como ainda não se tem computadores quânticos robustos à disposição, as simulações se fazem necessárias para antecipar a execução e os resultados dos algoritmos. O software simulador foi construído com base no formalismo matemático, tendo sido simulados vários outros algoritmos-base (tais como a busca de Grover, a estimativa de fase e a contagem quântica), com resultados que se mostraram coerentes com as predições da teoria, fundamentando por sua vez os algoritmos mais complexos elaborados aqui.

O uso de **casos práticos** para a demonstração dos algoritmos também foi feito, fornecendo uma ideia de como tais algoritmos podem auxiliar nas tarefas relacionadas a ontologias que foram tratadas aqui. Ainda que os exemplos tenham lidado com ontologias simples, pode-se inferir o uso para ontologias mais complexas, relativas ainda a outros tipos de domínio de conhecimento diferentes dos abordados aqui. Uma preocupação ao final de cada caso apresentado foi a análise dos resultados do algoritmo em relação à abordagem clássica usual.

Também ficou evidente no trabalho que, para se utilizar os algoritmos quânticos, um modo dual clássico-quântico de trabalho teve de ser idealizado, sendo que os dados deveriam "transitar" entre os dois modos dos algoritmos. Isto mostrou a necessidade de uma memória quântica que guardasse a superposição de instâncias ou classes, diferente de uma memória clássica. Assim, a ontologia deveria ser armazenada **temporariamente** em um modo quântico para o devido processamento. A evolução desta ideia de um armazenamento **temporário** para um modo **permanente** consistiu a base para o desenvolvimento da parte referente à engenharia de ontologias. Esta expansão não deveria apenas trazer os benefícios para os algoritmos quânticos, mas também quanto à própria forma de construção de ontologias, mostrando as possibilidades de se lidar com o consenso e formas conceituais e formais através de uma abordagem peculiar. Portanto, o *framework* Ontologias-CQ fica completo no sentido de haver tanto uma forma de uso de algoritmos quanto da própria construção de

ontologias em um modo clássico-quântico, caracterizando-se assim a sua potencial utilidade para a Engenharia do Conhecimento.

5.2 RECOMENDAÇÕES PARA TRABALHOS FUTUROS

Uma variedade de **trabalhos futuros** pode ser visualizada em função da convergência das áreas de conhecimento envolvidas:

1) Algoritmos clássico-quânticos para outras tarefas relacionadas a ontologias;
2) Estudos mais aprofundados em relação à Informação Quântica e uso de algoritmos quânticos de correção de erro;
3) Os algoritmos quânticos podem também ser desenvolvidos ou modificados com portas quânticas especificas de certos paradigmas de hardware quântico (computação quântica óptica, RMN, armadilhas iônicas, etc);
4) Estudos comparativos com abordagens de lógica difusa ou probabilísticas de processamento de informação;
5) Criação ou modificação de linguagens de ontologias com as características de construção ou processamento quântico;
6) Uso de formas alternativas da abordagem quântica, tal como a Computação Quântica Adiabática;
7) Como os algoritmos seguem as propriedades da computação reversível, podem ser implementados em outros paradigmas de hardware que funcionem sob estas propriedades.

A comparação da **complexidade** dos algoritmos ficou restrita aqui à prevalência do algoritmo ou sub-rotina quântica de menor complexidade em relação ao algoritmo clássico, sendo predominante $O(\sqrt{N/M})$ da contagem quântica em relação a $O(N)$ ou $O(N^2)$ dos algoritmos clássicos. Estudos mais aprofundados podem ser empreendidos para maior definição da vantagem em termos de complexidade algorítmica de tempo ou de espaço.

Em relação ao tema do processamento estocástico, a superposição de classes ou conceitos, utilizada como forma de lidar com a percepção difusa em uma representação de conhecimento, parece apresentar certo grau de isomorfismo com a lógica difusa ou *fuzzy*. O uso da inferência com lógica *fuzzy* envolve a construção de um modelo onde duas variáveis têm uma região de interpretação difusa, devendo ser simulado num computador clássico para se contornar o problema da lógica clássica. No entanto, uma ontologia em regime clássico-quântico tem embutida na sua construção uma **forma natural** de manipulação de

elementos em superposição, com a possibilidade de ser processada diretamente num computador quântico.

No intuito de facilitar a modelagem de ontologias em modo clássico-quântico, a **linguagem de representação de axiomas** deve ser modificada para contemplar as propriedades da superposição e emaranhamento, avançando além da lógica de primeira ordem e seguindo-se então a lógica quântica. O código do axioma deve informar se a validação ou a inferência será feita em algoritmo quântico. Tais modificações podem ser inspiradas nas propostas de linguagens que lidam com a lógica quântica, tais como QHaskell (VIZZOTO e COSTA, 2005) ou QCL (ÖMER, 2005).

Finalmente, os algoritmos quânticos contidos no *framework* foram apresentados de maneira formal e conceitual, bem como exemplos (da mesma forma que os "testes de mesa" dos algoritmos clássicos) foram associados a cada um, mostrando a viabilidade teórica do funcionamento dos algoritmos. Porém, fundamentando-se o paradigma quântico associado às Engenharia Ontológica exclusivamente sobre a Computação Quântica, garante-se a factibilidade da mesma como recurso e ferramenta para a Engenharia do Conhecimento. Mantendo-se esta perspectiva, o surgimento de computadores quânticos mais robustos (ver Apêndice A) permitirá a **demonstração prática** de tais algoritmos.

REFERÊNCIAS

ADAMOWSKI,; J. BEDNAREK,; S. SZAFRAN, B. Quantum Computing with Quantum Dots. **Schedae Informaticae**, v.14, p.95-111, 2005.

AHARONOV, D. Quantum Computation. In: STAUFFER, D. (Ed) **Annual Reviews of Computational Physics VI**, World Scientific, 1998.

_____.; VAN DAM, W.; KEMPE, J.; LANDAU, Z.; LLOYD, S.; REGEV, O. Adiabatic Quantum Computation is Equivalent to Standard Quantum Computation. **SIAM Journal of Computing**, v. 37, n.1, p. 166-194, 2007.

ALBERT, D. Z.; GALCHEN, R. Was Einstein Wrong? A Quantum Threat to Special Relativity. **Scientific American**, v.300, n.32, p.32-41, mar. 2009.

ALBERTI, A.; IVANOV, V. V.; TINO, G. M.; FERRARI, G. Engineering the quantum transport of atomic wavefunctions over macroscopic distances. **Nature Physics**, v.5, p.547-550, 2009.

ALTENKIRCH, T.; GRATTAGE, J. A functional quantum programming language**.** In: **LICS 2005 - Logic in Computer Science**. Proceedings of the 20th Annual IEEE Symposium, 26-29 jun. 2005, p.249-258.

ALTMAN, C. Quantum State Engineering with the rf-SQUID: A Brief Introduction. **NATO Advanced Research Workshop on Quantum Chaos**, 2003. Disponível em: <quant-ph/0307101>. Acesso em: 30/06/2010.

ANDRADE, A. L.; SELEME, A.; RODRIGUES, L. H.; SOUTO, R. **Pensamento Sistêmico – Caderno de Campo.** Porto Alegre: Ed. Bookman, 2006.

AOKI, T.; TAKAHASHI, G.; KAJIYA, T.; YOSHIKAWA, J.; BRAUNSTEIN, S. L.; VAN LOOCK, P.; FURUSAWA, A. Quantum error correction beyond qubits**. Nature Physics**, v.5, p.541-546, 2009.

ARPÍREZ, J.C.; CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A. WebODE in a Nutshell. **AI Magazine**, v.24, n.3, fall 2003. Disponível em: <http://www.aaai.org/ojs/index.php/aimagazine/article/view/1717/1615>. Acesso em: 30/06/2010.

_____.; GÓMEZ-PÉREZ, A.; LOZANO, A.; PINTO, S.H. (ONTO)2Agent: an Ontology-based WWW broker to Select Ontologies. **Workshop on Application of Ontologies and Problems Solving Methods**. ECAI'98. Brighton, 1998.

ASPECT, A.; GRANGIER, P.; ROGER, G. Experimental Realization of Einstein-Podolsky-Rosen-Bohm Gedankenexperiment: A New Violation of Bell´s Inequalities. **Physical Review Letters**, v.49, n.2, p.91-94, 1982.

AWSCHALOM, D. D.; FLATTÉ, M. E.; SAMARTH, N. The Diamond Age of Spintronics. **Scientific American**, v.297, n.84, jun. 2002.

_____.; EPSTEIN, R.; HANSON, R. **A Idade dos Diamantes na Spintrônica**. Scientific American Brasil, n.66, 2007.

BAADER, F.; HOLLUNDER, B. KRIS: Knowledge Representation and Inference System. **ACM SIGART Bulletin** v.2, n.3, p.8-14, 1991.

_____.; CALVANESE, D.; MCGUINNESS, D. L.; NARDI, D.; PATEL-SCHNEIDER, P. F. (Eds.) **The Description Logic Handbook: Theory, Implementation, and Applications**. United Kingdom: Cambridge University Press, 2003.

BACON, D.; LEUNG, D. Toward a World with Quantum Computers. **Communications of the ACM**, v.50, n.9, p.55-59, set. 2007.

BARBOSA, A. A.; LULA JR, B.; LIMA, A. F. Uma ferramenta de simulação numérica e simbólica de circuitos quânticos. In: **WEICQ 2006: Workshop-Escola de Computação e Informação Quântica**, out. 2006.

BARENCO, A; BENNETT, C. H.; CLEVE, R.; DIVINCENZO, D. P.; MARGOLUS, N.; SHOR, P.; SLEATOR, T.; SMOLIN, J. A.; WEINFURTER, H. Elementary gates for quantum computation, **Physics Review A** v.52, n.5, p.3457–3467, 1995.

BARRETT, M. D.; CHIAVERINI, J.; SCHAETZ, T.; BRITTON, J.; ITANO, W. M.; JOST, J. D.; KNILL, E.; LANGER, C.; LEIBFRIED, D.; OZERI, R.; WINELAND, D. J. Deterministic quantum teleportation of atomic qubits. **Nature**, n.429, p.737-739, 2004.

BASSILIADES, N.; ANTONIOU, G.; VLAHAVAS, I. A Defeasible Logic Reasoner for the Semantic Web. **LNCS-Lecture Notes in Computer Science**, Berlin/Heidelberg, Springer-Verlag, v.3323, p.49-64, 2004.

BELL, J. S. On the Einstein-Podoslky-Rosen paradox**.** Physics, n.1, p.195-200, 1964. Reimpresso em BELL, J. S. **Speakble and Unspeakble in Quantum Mechanics**. Cambridge University Press, 1987.

BENNETT, C. H. Logical reversibility of computation, **IBM Journal of Research and Development**, v.17, p.525-532, 1973.

_____.; BRASSARD, G.; Quantum Criptography: Public Key Distribution and Coin Tossing**.** In: IEEE International Conference on Computers, Systems and Signal Processing, New York, 1984. **Proceedings…** Bangalore, India: IEEE, dez.1984, p.175-179.

_____.; BRASSARD, G.; CRÉPEAU, C.; JOSZA, R.; PERES, A.; WOOTTERS, W. K. Teleporting an Unknown Quantum State via Dual Classical and EPR Channels. **Physical Review Letters**, v.70, p.1895-1899, 1993.

BENIOFF, P. The computer as a physical system: A microscopic quantum mechanical Hamiltonian model of computers as represented by Turing machines. **Journal of Statistical Physics**, v.22, n.5, p.563-591, 1980.

BERNERS-LEE, T.; HENDLER, J.; LASSILA, O. **The Semantic Web**. Scientific American, v.284, n.5, p.35-43, 2001.

BERNSTEIN E; VAZIRANI, U. Quantum complexity theory. In: 25th Annual ACM Symposium on Theory of Computing. **Proceedings…** New York: ACM, 1993, p.11-20.

BERTHIAUME, A; BRASSARD, G. The quantum challenge to structural complexity theory. In: 7th Annual Structure in Complexity Theory Conference. **Proceedings…** Los Alamitos, CA: IEEE Computer Society Press, 1992, p.132-137.

BIALCZAK, R. C.; ANSMANN, M.; HOFHEINZ, M.; LUCERO, E.; NEELEY, M.; O'CONNELL, A. D.; SANK, D.; WANG, H.; WENNER, J.; STEFFEN, M.; CLELAND, A. N.; MARTINIS, J. M. Quantum process tomography of a universal entangling gate implemented with Josephson phase qubits. **Nature Physics**, n.6, p.409-413, 2010.

BORST, W. N. **Construction of engineering ontologies,** Enschede, Netherlands, 1997, 227f. Tese (Doutorado)-Universiteit Twente. Disponível em: <http://www.ub.utwente.nl/webdocs/inf/1/t0000004.pdf>. Acesso em 30/06/2010.

BOUWMEESTER, D.; PAN, J.W.; DANIELL, M.; WEINGURTER, H.; ZEILINGER, A. **Observation of Three-Photon GHZ Entanglement**. Physics Review Letters, v.82, n.7, p.1345-1349, 1999.

BRASSARD, G.; HOYER, P.; TAPP, A. **Quantum Counting**. http://arXiv.org/abs/quant-ph/9805082v1, 1998. Acesso em 30/06/2010.

BRUZA, P.; KITTO, K.; NELSON, D.; McEVOY, C. Extracting Spooky-activation-at-a-distance from Considerations of Entanglement. **LNCS-Lecture Notes in Computer Science**, Berlin/Heidelberg: Springer-Verlag, v.5494, p.71-83, 2009.

BURKARD, G.; LOSS, D.; DIVINCENZO, D. P. **Coupled quantum dots as quantum gates**. Physics Review B, v.59, n.3, p. 2070-2080, 1999.

CABRAL, G. E. M.; LULA, B.; LIMA, A. F. ZENO: a new graphical tool for design and simulation of quantum circuits. **Quantum Information and Computation III**, v. 5815, pages 127–137. SPIE, 2005.

CALDERBANK, A. R.; SHOR, P. W. Good quantum error-correcting codes exist. **Physical Review A**, v.54, n.2, p.1098-1105, 1996.

CARUSO, F.; OGURI, V. **Física Moderna – Origens Clássicas e Fundamentos Quânticos**. Rio de Janeiro: Ed. Campus, 2006.

CHRISLEY, R. L. Quantum learning. In: PYLKKÄNEN, P.; PYLKKÖ, P. (Eds.) New Directions in Cognitive Science: International Symposium, Saariselka, 4-9 de agosto de 1995, **Proceedings…** Lapland, Finland: Finnish Association of Artificial Intelligence, 1995, p.77-89.

CIRAC, J.; ZOLLER, P. Quantum Computations with Cold Trapped Ions. **Physical Review Letters**, v.74, n.20, p.4091-4094, 1995.

COLLINS, G. Nós Quânticos na Computação. **Scientific American Brasil**, n.4, p.49-55, 2006.

CORCHO, O.; FERNÁNDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A.; VICENTE, O. WebODE: An Integrated Workbench for Ontology Representation, Reasoning, and Exchange. In: GÓMEZ-PÉREZ, A.; BENJAMINS, V. R. (Eds). **Lecture Notes in Artificial Intelligence LNAI**. Berlin/Heidelberg: Springer-Verlag, v.2473, p.138-153, 2002. Disponível em <http://www.springerlink.com/content/vceph8xydcdw1gae/>. Acesso em 30/06/2010.

CORMEN, T.H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. **Algoritmos: Teoria e Prática**. 2ªed. Rio de Janeiro: Campus, 2002.

CULLIMORE, J. 'Fab Club' Announces High-K 28nm Chip Technology. **ITPro Portal**, 15 de junho de 2010. Disponível em <http://www.itproportal.com/portal/news/article/2010/6/15/fab-club-announces-high-k-28nm-chip-technology/>. Acesso em 30/06/2010.

DEAN, M.; SCHREIBER, G. OWL Web Ontology Language Reference. **W3C Working Draft**. 2003. Disponível em <http://www.w3.org/TR/owl-ref/>. Acesso em 30/06/2010.

DEMARCO, B.; BEN-KISH, A.; LEIBFRIED, D.; MEYER, V.; ROWE, M.; JELENKOVIC, B. M.; ITANO, W. M.; BRITTON, J.; LANGER, C.; ROSENBAND, T.; WINELAND, D. J. Experimental demonstration of a controlled-NOT wave-packet gate**. Physics Review Letters**, v.89, n.26, p.267901-1 - 267901-4, 2002.

DE RAEDT, H.; HAMS, A. H.; MICHIELSEN, K.; DE RAEDT, K. Quantum Computer Emulator. **Computer Physics Communications** v.132, p.1–20, 2000.

DE ROO, J. OWL implementation experience in Euler (palestra). 5th Meeting of the W3C Web Ontology Working Group, Manchester, UK, 9-10 jan. 2003. **Proceedings...** Disponível em <http://www.agfa.com/w3c/Talks/2003/01webont/Overview.html>. Acesso em 30/06/2010.

DE VRIES, A**. jQuantum – A Quantum Computer Simulator Version 0.9b Documentation,** 2006. Disponível em <http://jquantum.sourceforge.net/jQuantum.pdf>. Acesso em 08/08/2008.

DEUTSCH, D. It From Qubit. In: BARROW, J.; DAVIES, P.; HARPER, C. (Eds). **Science & Ultimate Reality**. Cambridge, UK: Cambridge University Press, 2003.

_____.; JOZSA, R. Rapid solutions of problems by quantum computation. **Proceedings of the Royal Society of London A**, v. 439, n.1907, p.553-558, 1992.

_____. Quantum Theory, the Church-Turing principle and the universal quantum computer, **Proceedings of the Royal Society of London A**, v.400, n.1818, p.97-117, 1985.

DICARLO, L.; CHOW, J. M.; GAMBETTA, J. M.; BISHOP, L. S.; JOHNSON, B. R.; SCHUSTER, D. I.; MAJER, J.; BLAIS, A.; FRUNZIO, L.; GIRVIN, S. M.; SCHOELKOPF, R. J. Demonstration of

two-qubit algorithms with a superconducting quantum processor. **Nature Physics**, v.460, n.7252, p.240-244, 28 jun.2009.

DIEKS, D. Communication by EPR devices**. Physical Review Letters A**, v.92, n.6, p.271-272, 1982.

DIVICENZO, D. P. Topics in Quantum Computers. In: KOWENHOVEN, L., SCHÖN, G. & SOHN, L. (Eds). Mesoscopic Electron Transport**. NATO Advanced Study Institute,** Series E. Dordrecht : Kluwer Ac. Publ., v.345, 1997. Disponível em <cond-mat/9612126>. Acesso em 30/06/2010.

DODIG-CRNKOVIC, G. **Scientific Methods in Computer Science**. Conference for the Promotion of Research in IT at New Universities, 2002, Sweden. **Anais…** [s.l] University Colleges in Sweden.

DONG, D. Y.; CHEN, C. L.; LI, H.; TARN, T. Quantum Reinforcement Learning, **IEEE Transactions on Systems, Man and Cybernetics – Part B**: **Cybernetics**, v.38, n.5, p.1207-1220, out. 2008.

_____.; CHEN, Z. H.; ZHANG, C. B. Quantum mechanics helps in learning for more intelligent robots. **Chinese Physics Letters**, v.23, n.7, p.1691-1694, jul. 2006.

_____.; ZHANG, C. B.; CHEN, Z. H. Quantum robot: structure, algorithms and applications. **Robotics**, v.24, n.4, p.513-521, jul. 2006.

DOUSSE, A.; SUFFCZYSKI, J.; BEVERATOS, A.; KREBS, O.; LEMAITRE, A.; SAGNES, I.; BLOCH, J.; VOISIN, P.; SENELLART, P. Ultrabright source of entangled photon pairs**. Nature**, v.466, n.7303, p.217-220, jul. 2010.

ECCLES, J. C. A unitary hypothesis of mind-brain interaction in the cerebral cortex. **Proceedings of the Royal Society of London B,** v.240, n.1299, p.433-451, 1990.

EKERT, A.; HAYDEN, P.; INAMORI, H. **Basic concepts in quantum computation**, 2000. Disponível em <http://arxiv.org/pdf/quant-ph/0011013v1>. Acesso em 30/06/2010.

EISBERG, R.; RESNICK, R. **Física Quântica**. Rio de Janeiro: Ed. Campus, 1979. Reimpressão.

EINSTEIN, A.; PODOLSKY, B.; ROSEN, N. Can Quantum-Mechanical Description of Physical Reality Be Considered Complete? **Physical Review**, v.47, n.10, p.777-780, 1935.

EZHOV, A.; VENTURA, D. Quantum Neural Networks. In: KASABOV, N (ed). **Future Directions for Intelligent Systems and Information Sciences**. Heidelberg: Physica-Verlag, 2000, 420p.

FARHI, E.; GOLDSTONE, J.; GUTMANN, S.; SIPSER, M. **Quantum computation by adiabatic evolution,** 2000. Disponível em <arXiv:quant-ph/0001106>. Acesso em 30/06/2010.

FÉRNANDEZ-LÓPEZ, M.; GÓMEZ-PÉREZ, A.; JURISTO, N. METHONTOLOGY: From Ontological Art Toward Ontological Engineering. **Spring Symposium on Ontological Engineering of AAAI**. 1997, California, Stanford University, p.33-40.

FEYNMAN, R. Simulating Physics with Computers. **International Journal of Theoretical Physics**, v.21, n.6/7, p.467-488, 1982.

FIKES, R.; JENKINS, J.; FRANK, G. JTP: A System Architecture and Component Library for Hybrid Reasoning. 7th Seventh World Multiconference on Systemics, Cybernetics, and Informatics. Orlando, Florida, USA. **Proceedings…** jul. 2003.

FREDKIN, E.; TOFFOLI, T. Conservative logic. **International Journal of Theoretical Physics**, v.21, p.461-488, 1982.

FREEDMAN, S. J.; CLAUSER, J. F. Experimental test of local hidden-variable theories. **Physics Review Letters**, v.28, n.14, p.938-941, 1972.

FUECHSLE, M.; MAHAPATRA, S.; ZWANENBURG, F. A.; FRIESEN, M.; ERIKSSON, M. A.; SIMMONS, M. Y. Spectroscopy of few-electron single-crystal silicon quantum dots**. Nature Nanotechnology**, v.5, p.502-505, jul.2010.

GAO, W.; LU, C.; YAO, X.; XU, P.; GÜHNE, O.; GOEBEL, A.; CHEN, Y.; PENG, C.; CHEN, Z.; PAN, J. Experimental demonstration of a hyper-entangled ten-qubit Schrödinger cat state. **Nature Physics**, n.6, p.331-335, mar. 2010.

GIOVANNETTI, V.; LLOYD, S.; MACCONE, L. Quantum random access memory. **Physics Review Letters**, v.100, n.16, p.160501, 4p. abr. 2008. (a)

_____. Architectures for a quantum random access memory. **Physics Review A**, v.78, n.5, p.052310, 9p. nov. 2008. (b)

GÓMEZ-PÉREZ, A.; FERNANDEZ-LOPEZ, M.; CORCHO, O. **Ontological Engineering.** London: Springer-Verlag, 2004.

_____.; CORCHO. O. Ontology Languages for the Semantic Web**. IEEE Intelligent Systems**, v.17, n.1, p.54-60, jan./fev. 2002.

_____. Knowledge Sharing and Reuse. In: LIEBOWITZ, J. (ed) **Handbook of Expert Systems,** CRC 10. Boca Raton, Florida, 1998.

GOSWAMI, A.; REED, R. E.; GOSWAMI, M. **The Self-Aware Universe: How consciousness creates the material world**. New York: Most Tarcher/Putnam, 1993.

GREILICH, A.; ECONOMOU, S. E.; SPATZEK, S.; YAKOVLEV, D. R.; REUTER, D.; WIECK, A. D.; REINECKE, T. L.; BAYER, M. Ultrafast optical rotations of electron spins in quantum dots. **Nature Physics**, n.5, p.262-266, mar. 2009.

GROVER, L. K. A fast quantum-mechanical algorithm for database search. In: 28th Annual ACM Symposium on the Theory of Computing, Philadelphia PA, mai.1996, **Proceedings…** 1996**,** p.212-219.

GRUBER, T. R. A Translation Approach to Portable Ontology Specifications. **Knowledge Acquisition**, v.5, n.2, p.199-220, 1993(a).

Disponível em <http://ksl-web.stanford.edu/KSL_Abstracts/KSL-92-71.html>. Acesso em 30/06/2010.

_____. Toward principles for the design of ontologies used for knowledge sharing. In GUARINO, N.; R. POLI, R. (Eds.), **International Workshop on Formal Ontology**, Padova, Italy. Revised August 1993 (b). Disponível em <http://ksl-web.stanford.edu/KSL_Abstracts/KSL-93-04.html>. Acesso em 30/06/2010.

_____. **Ontology.** In: LIU, L.; ÖZSU**,** T. (Eds.) **Entry in the Encyclopedia of Database Systems**, Springer-Verlag, 2008.

GRÜNINGER, M.; FOX, M. S. **Methodology for the Design and Evaluation of Ontologies**. Em: SKUCE, D. (ed) IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, p.6.1-6.10, 1995.

GUALTIERI, A.; RUFFOLO, M. An Ontology-Based Framework for Representing Organizational Knowledge. **Proceedings of I-KNOW '05**. Graz, Austria, 2005.

GUARINO, N.; GIARETTA, P. Ontologies and Knowledge Bases: Towards a Terminological Clarification. In: MARS, N. (ed.) **Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing** (KBKS'95). University of Twente, Enschede, The Netherlands. IOS Press, Amsterdam, The Netherlands, 1995, p.25-32. Disponível em <http://www.loa-cnr.it/Papers/KBKS95.pdf>. Acesso em 30/06/2010.

_____. Formal Ontology in Information Systems. In: GUARINO, N. (Ed.) 1st International Conference on Formal Ontology in Information Systems (FOIS'98). Trento, Italy, 1998. **Proceedings…** IOS Press, Amsterdam, 1998, p.3-15. Disponível em : <http://citeseerx.ist.psu.edu/viewdoc/download;jsessionid=76AB7F0294EE598AE89CD2C0A949EE5D?doi=10.1.1.29.1776&rep=rep1&type=pdf>. Acesso em 30/06/2010.

HANNEKE, D.; HOME, J. P.; JOST, J. D.; AMINI, J. M.; LEIBFRIED, D.; WINELAND, D. J. Realization of a programmable two-qubit quantum processor**. Nature Physics**, n.6, p. 13-16, 2010.

HISCOCKS, M. P.; GANESAN, K.; GIBSON, B. C.; HUNTINGTON, S. T.; LADOUCEUR, F.; PRAWER, S. Diamond waveguides fabricated by reactive ion etching. **Optics Express**, v.16, n.24, p.19512-19519, 2008.

HOFFMAN, G. **CIE Color Space**. Department of Mechanical Engineering, University of Applied Sciences, Emden-Germany, 2000. Disponível em <http://www.fho-emden.de/~hoffmann/ciexyz29082000.pdf>. Acesso em 30/06/2010.

HOGG, T.; PORTNOV, D. Quantum Optimization, **Information Science**, v.128, n.3, p.181-197, out.2000.

_____. Quantum Computing and Phase Transitions in Combinatorial Search. **Journal of Artificial Intelligence Research**, v.4, p.91-128, mar. 1996.

HOME, J. P.; HANNEKE, D.; JOST, J. D.; AMINI, J. M.; LEIBFRIED, D.; WINELAND, D. J. Complete Methods Set for Scalable Ion Trap Quantum Information Processing. **Science Express**, v.325, n.5945, p.1227-1230, ago. 2009.

HONJO, T.; NAM, S. W.; TAKESUE, H.; ZHANG, Q.; KAMADA, H.; NISHIDA, Y.; TADANAGA, O.; ASOBE, M.; BAEK, B.; HADFIELD, R.; MIKI, S.; FUJIWARA, M.; SASAKI, M.; WANG, Z.;. INOUE, K; YAMAMOTO, Y. Long-distance entanglement-based quantum key distribution over optical fiber. **Optics Express,** v.16, p.19119-19126, nov. 2008.

HORROCKS, I., FENSEL, D., HARMELEN, F., DECKER, S., ERDMANN, M, KLEIN, M. OIL in a Nutshell. In: DIENG, R.; CORBY, O. (eds) 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00), Juan-Les-Pins, France. **Proceedings…** Berlin, Germany: Springer-Verlag, LNAI 1937, 2000, p.161-180.

_____.; VAN HARMELEN, F (Eds). **Reference Description of the DAML+OIL Ontology Markup Language**. Relatório Técnico, mar. 2001. Disponível em <http://www.daml.org/2001/03/reference.html>. Acesso em 30/06/2010.

HUANG, S. M.; TOKURA, Y.; AKIMOTO, H.; KONO, K.; LIN, J. J.; TARUCHA, S.; ONO, K. Spin Bottleneck in Resonant Tunneling through Double Quantum Dots with Different Zeeman Splittings. **Physical Review Letters**, v. 104, n.13, p.136801, 4p. abr. 2010.

JANG, M.; JOO-CHAN, S.; Bossam: An extended rule engine for OWL inferencing. International Workshop RuleML 2004 - Rules and rule markup languages for the semantic web, **Anais…** Hiroshima, Japão, nov. 2004, v.3323, p.128-138.

KAK, S. C. On Quantum Neural Computing. **Information Science**, v.83, n.3, p.143-160, mar. 1995.

KARVOUNARAKIS, G.; MAGGANARAKI, A.; ALEXAKI, S.; CHRISTOPHIDES, V.; PLEXOUSAKIS, D.; SCHOLL, M.; and TOLLE, K. Querying the Semantic Web with RQL. **Computer Networks**, v.42, n.5, p.617-640. ago. 2003. Disponível em <http://dx.doi.org/10.1016/S1389-1286(03)00227-5>. Acesso em 30/06/2010.

KNILL, E. **Conventions for Quantum Pseudocode**. LANL report LAUR-96-2724, 1996. Disponível em: <http://citeseerx.ist.psu.edu /viewdoc/summary?doi=10.1.1.57.1328>. Acesso em 30/06/2010.

KNOLL, Y. **Coupled wave-particle dynamics as the ontology behind Quantum Mechanics and long-range interactions,** mai. 2006. Disponível em: <http://arxiv.org/abs/quant-ph/0605011>. Acesso em 30/06/2010.

KOPENA, J. B.; REGLI, W. C. DAMLJessKB: A Tool for Reasoning with the Semantic Web. **The Semantic Web - ISWC 2003**, p.628-643, 2003.

KRAUSE, D. Remarks on Quantum Ontology. **Journal Synthese – Humanities, Social Sciences and Law Collection**. Springer Netherlands, v.125, n1-2, p.155-167, 2000.

KUZNETSOVA, E.; GACESCA, M.; YELIN, S. F.; CÔTÉ, R. Phase gate and readout with an atom-molecule hybrid platform. **Physical Review A**, v. 81, n.3, p.030301(R), 4p. mar. 2010.

LAFLAMME, R.; KNILL, E. ; CORY, D.; FORTUNATO, E. M.; HAVEL, T.; MIQUEL, C.; MARTINEZ, R.; NEGREVERGNE, C.; ORTIZ, G.; PRAVIA, M. A.; SINHA, S.; SOMMA, R. ; VIOLA, L. **Introduction to NMR quantum information processing**. Technical Report LAUR-02-6132, Los Alamos National Laboratory, 2001.

LENAT, D. B.; GUHA, R. V. **Building Large Knowledge-Based Systems: Representation and Inference in the Cyc Project**. Addison-Wesley, Boston, Massachusetts, 1990.

LEUENBERGER, M. N.; LOSS, D.; POGGIO, M.; AWSCHALOM, D. D. Quantum information processing with large nuclear spins in GaAs semiconductors. **Physics Review Letters**, v.89, n.20, p.207601, 4p. out.2002.

LOMONT, C. **Quantum convolution and quantum correlation algorithms are physically impossible,** 2003. Disponível em <arXiv:quant-ph/0309070v2>. Acesso em 30/06/2010.

LOSS D.; DIVINCENZO D.P. Quantum Computation with Quantum Dots. **Physics Review A**, v.57, n.1, p.120-127, jan. 1998.

LOTH, S.; VON BERGMANN, K.; TERNES, M.; OTTE, A. F.; LUTZ, C. P.; HEINRICH, A. J. Controlling the state of quantum spins with electric currents. **Nature Physics**, n.6, p.340-344, 2010.

MACGREGOR, R. Inside the LOOM Classifier. **SIGART Bulletin**, v.2, n.3, p.70-76, 1991.

MAEDCHE, A; STAAB, S. Semi-automatic Enginnering for Ontologies from Text. In: CHANG, S. K.; OBOZINSKI, W. R. (Eds) SEKE 2000 - 12th International Conference on Software Engineering and Knowledge Engineering, **Anais…** Chicago, Illinois, 2000.

MALOSSINI, A.; BLANZIERI, E.; CALARCO, T. Quantum Genetic Optimization. **IEEE Transactions on Evolutionary Computation**, v.12, n.2, p.231-241, abr. 2008.

McCAMEY, D. R.; MORLEY, G. W.; SEIPEL, H. A.; BRUNEL, L. C.; van TOL, J.; BOEHME, C. Spin-dependent processes at the crystalline Si-SiO2 interface at high magnetic fields. **Physical Review B**, v.78, n.4, p.045303, 5p. nov. 2008.

McGUINNESS, D. L.; da SILVA, P. P. Explaining answers from the Semantic Web: the Inference Web approach**. Web Semantics: Science, Services and Agents on the World Wide Web,** v.1, p.397–413, 2004.

MEDEIROS, L. F.; BASTOS, L. C.; BASTOS, R. C. **Um Algoritmo de Computação Quântica para Merging de Ontologias**. 3º ONTOBRÁS – Seminário de Pesquisa em Ontologias no Brasil, ago. 2010, Florianópolis-SC.

_____.; RAUTENBERG, S.; IGARASHI, W.; IGARASHI, D. C. C. Ferramentas de Engenharia do Conhecimento como Suporte ao Processo de Aprendizagem Organizacional. In: IV Congresso da Academia Trinacional de Ciências, 2009, Foz do Iguaçu. **Anais**..., ISSN: 1982-2758, 2009.

_____.; _____.; _____.; FIALHO, F. A. P.; SANTOS, N.; BASTOS, R. C. Um Modelo de Memória de Trabalho Artificial Utilizando Ontologias. ABERGO 2006, 14º Congresso Brasileiro de Ergonomia, out./nov. 2006, **Anais...** Curitiba-PR, 2006.

MIZOGUCHI, R.; VANWELKENHUYSEN, J.; IKEDA, M. Task Ontology for Reuse of Problem Solving Knowledge. In: MARS, N. (Ed) **Towards Very Large Knowledge Bases: Knowledge Building and Knowledge Sharing**. University of Twente, Enschede, Holanda. IOS Press, Amsterdam, 1995, p.46-57.

MOORE, G. Cramming more components onto integrated circuits. **Electronics**, v.38, n.8, abr. 1965.

MORIN, E. **Introdução ao Pensamento Complexo**. Porto Alegre: Ed. Sulina, 2005.

MOTTA, E. **Reusable Components for Knowledge Modelling: Principles ans Case Studies in Parametric Design**. The Netherlands: IOS Press. Amsterdam, 1999.

NAGY, M.; AKL, S. G. **Quantum Computation and Quantum Information**. Technical Report No. 2005-496, School of Computing, Queen´s University, Kingston, Ontario, K7L 3N6, Canada, 2005.

NECHES, R.; FIKES, R. E.; FININ, T.; GRUBER, T. R.; SENATOR, T,. SWARTOUT, W. R. Enabling technology for knowledge sharing. **AI Magazine,** v.12, n.3, p:36-56, 1991. Disponível em: <http://tomgruber.org/writing/AIMag12-03-004.pdf>. Acesso em: 30/06/2010.

NEGREVERGNE, C.; MAHESH, T. S.; RYAN, C. A.; DITTY, M.; CYR-RACINE, F.; POWER, W.; BOULANT, N.; HAVEL, T.; CORY, D. G.; LAFLAMME, R. Benchmarking quantum control methods on a 12-qubit system. **Physics Review Letters**, v.96, n.17, p.170501, mai. 2006.

NIELSEN, M. A.; CHUANG, I. L. **Computação Quântica e Informação Quântica.** Porto Alegre: Ed. Bookman, 2005.

NOY, N. F.; FERGERSON, R. W.; MUSEN, M. A. The knowledge model of Protege-2000: Combining interoperability and flexibility. In: DIENG, R.; CORBY, O. (eds) 12th International Conference in Knowledge Engineering and Knowledge Management (EKAW'00). Juan-Les-Pins, France. **Proceedings…** Berlin, Germany: LNAI 1937, Springer-Verlag, 2000, p.17-32. Disponível em: <http://www.pms.ifi.lmu.de/mitarbeiter/ohlbach/Ontology/Protege/SMI-2000-0830.pdf> . Acesso em 30/06/2010.

_____.; MUSEN, M. A. SMART: Automated Support for Ontology Merging and Alignment. In: GAINES, B. R.; KREMER, B.; MUSEN, M. A. (eds) 12th Banff Workshop on Knowledge Acquisition, Modeling and Management. **Anais…** Banff, Alberta, Canada, 4-7:1-20, 1999.

OLMSCHENK, S.; MATSUKEVICH, D. N.; MAUNZ, P.; HAYES, D.; DUAN, L.-M.; MONROE, C. Quantum teleportation between distant matter qubits. **Science**, v.323, n.5913, p.486-489, jan. 2009.

ÖMER, B. **A procedural formalism for quantum computing**. 1998. Tese de Mestrado. Department of Theoretical Physics, Technical University of Vienna.

_____. **Structured Quantum Programming**. Institute for Theoretical Physics, Vienna University of Technology, 2003. Disponível em: <http://www.itp.tuwien.ac.at/~oemer/>. Acesso em 08/08/2009.

OLIVEIRA, I. S.; VIEIRA, C. L. **A Revolução dos Q-bits: o Admirável Mundo da Computação Quântica**. Rio de Janeiro: Zahar, 2009.

PATEL, A. **What is Quantum Computation?** Report Number CTS-IISc/6-99, CTS and SERC, Indian Institute of Science, Bangalore, 1999. Disponível em: <quant-ph/9909082v1>. Acesso em 30/06/2010.

PENROSE, R. **A Mente Nova do Rei**. Rio de Janeiro: Campus, 1991.

PERRY, R. T. **The Temple of Quantum Computing**, 250p, 2006. Disponível em: <http://www.toqc.com/TOQCv1_1.pdf>. Acesso em 30/06/2010.

PERSEGUERS, S.; LEWENSTEIN, M.; ACÍN, A.; CIRAC, J. I. Quantum Random Networks. **Nature Physics**, v.6, p.539-543, mai. 2010.

PESSOA JÚNIOR, O. **Conceitos de Física Quântica**. São Paulo: Ed. Livraria da Física, v.1, 2003.

_____. **Conceitos de Física Quântica**. São Paulo: Ed. Livraria da Física, v.2, 2006.

POPPER, K. R.; ECCLES, J. C. **The Self and its Brain**. Berlin: Springer-Verlag International, 1977.

POLANYI, M. **The Tacit Dimension**. Londres: Routledge e Kegan Paul, 1966.

PORTUGAL, R.; LAVOR, C. C.; CARVALHO, L. M.;MACULAN, N. Uma Introdução à Computação Quântica. **Notas em Matemática Aplicada**, v.8. São Carlos, SP: SBMAC, 2004.

PRESKILL, J. **Lecture Notes for Physics 229: Quantum Information and Computation**. California Institute of Technology, 1998. Disponível em: <http://www.theory.caltech.edu/people/preskill/ph229/>. Acesso em: 30/06/2010.

RAUTENBERG, S. ; TODESCO, J. L. ; GAUTHIER, F. O. Processo de desenvolvimento de ontologias: uma proposta e uma ferramenta. **Revista Tecnologia (UNIFOR)**, v. 30, p. 133-144, 2009.

REED, S. L.; LENAT, D. B. Mapping Ontologies into Cyc. In: AAAI 2002 Conference Workshop on Ontologies for the Web Semantic, **Anais…** Edmonton, Canada, jul. 2002.

REID, T. **On the Evolutionary Design of Quantum Circuits**, 2005. Tese de Mestrado, University of Waterloo, Ontario, Canada.

RIEFFEL, E.; POLAK, W. An Introduction to Quantum Computing for Non-Physicists. **ACM Computing Surveys**, v.32, n.3, p.300-335, set. 2000. Disponível em: <http://ru.arxiv.org/abs/quant-ph/9809016v2>. Acesso em 30/06/2010.

RIGATOS, G. G.; TZAFESTAS, S. G. Parallelization of a fuzzy control algorithm using quantum computation. **IEEE Transactions on Fuzzy Systems**, v.10, n.4, p.451-460, ago. 2000.

ROSS, M. Quantum Computing. **Communications of the ACM**, v.51, n.7, p.12-13, jul.2008.

RUBINSTEIN, B. Evolving quantum circuits using genetic programming. In: KOZA, J. R. (Ed.). **Genetic Algorithms and Genetic Programming at Stanford.** Stanford University-CA, Stanford Bookstore, 2000, p.325-334.

RUSSEL, S. & NORVIG, P. **Inteligência Artificial**, 2ª ed. Rio de Janeiro: Campus-Elsevier, 2004.

SATINOVER, J. **O Cérebro Quântico: As Novas Descobertas da Neurociência e a Próxima Geração de Seres Humanos**. São Paulo: Ed. Aleph, 2007.

SCHAETZ, T.; BARRETT, M. D.; LEIBFRIED, D.; CHIAVERINI, J.; BRITTON, J.; ITANO, W. M.; JOST, J. D.; LANGER, C.; WINELAND, D. J**.** Quantum dense coding with atomic qubits. **Physics Review Letters**, v.93, n.4, p.040505-1 - 040505-4, 2004.

SCHREIBER, G.; AKKERMANS, H.; ANJEWIERDEN, A.; HOOG, R.; SHADBOLT, N.; DE VELDE, W. V.; WIELINGA, B. **Knowledge Engineering and Management: the CommonKADS Methodology**. Cambridge. Massachussets: MIT Press, 2002.

_____.; WIELINGA, B. J.; JANSWEIJER, W. The KACTUS View on the 'O' Word. Em: SKUCE,, D. (Ed) IJCAI95 Workshop on Basic Ontological Issues in Knowledge Sharing, p.15.1-15.10, 1995. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.1.5032&rep=rep1&type=pdf>. Acesso em 30/06/2010.

SELINGER, P. Towards a quantum programming language. **Mathematical Structures in Computer Science**, v.14, n.4, p.527–586, 2004.

SHOR, P. Algorithms for Quantum Computation: Discrete Logarithms and Factoring**.** In: **Proceedings of 35th Annual Symposium of Fundamentals of Computing Science**, p.124-134, 1994.

SIMON, D. R. On the power of quantum computation. In: **Proceedings of 35th Annual Symposium on Foundations of Computer Science**, IEEE Computer Society Press, Los Alamitos 1994, p.124-134.

STAAB, S.; SCHNURR, H. P.; STUDER, R.; SURE, Y. Knowledge Processes and Ontologies**. IEEE Intelligent Systems**, v.16, n.1, p.26-34, 2001.

STEANE, A. M. **Quantum computing,** 1997. Disponível em: <http://arxiv:quant-ph/9708022v2.pdf>. Acesso em: 30/06/2010.

_____. Error correcting codes in quantum theory. **Physical Review Letters**, v.77, n.5, p.793-797, 1996.

STUDER, R.; BENJAMINS, V. R.; FENSEL, D. Knowledge Engineering: Principles and Methods. **IEEE Transactions on Data and Knowledge Engineering**, v.25, n.1-2, p.161-197, 1998. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.45.4555&rep=rep1&type=pdf>. Acesso em: 30/06/2010.

_____; DECKER, S; FENSEL, D; STAAB, S. Situation and Perspective of Knowledge Engineering. In: CUENA, J. et al. (eds.), **Knowledge Engineering and Agent Technology**. The Netherlands/Amsterdam: IOS Press, 2000.

SWARTOUT, B.; RAMESH, P.; KNIGHT, K.; RUSS, T. Toward distributed use of large-scale ontologies. In: FARQUHAR, A.; GRÜNINGER, M.; GÓMEZ-PÉREZ, A.; USCHOLD, M.; VAN DER VET, P.; AAAI 97 Spring Symposium Series Workshop On Ontological Engineering. **Proceedings…** AAAI Press, p.138-148, 1997

TENENBAUM, A. M.; LANGSAM, Y.; AUGENSTEIN, M. J. **Estrutura de Dados usando C**. São Paulo: Makron Books, 1995.

TEJADA, J.; CHUDNOVSKY, N. D.; DEL BARCO, E.; HERNANDEZ, J. M.; SPILLER, T. P. **Magnetic Qubits as Hardware for Quantum Computers**. Trusted E-Services Laboratory, HP Laboratories Bristol. Tech Report HPL-2000-87, jul.2000.

THIERAUF, R.; KLEKAMP, R. **Decision making through operations research**. Nova York, EUA: John Willey and Sons, Inc., 1975.

TIJERINO, Y. A.; MIZOGUCHI, R. MULTIS II: Enabling End-users to Design Problem-Solving Engines via Two-Level Task Ontologies. In: AUSSENAC, N.; BOY, G.; GAINES, B.; LINSTER, M.; GANASCIA, J. G.; KODRATOFF, Y (eds). 7th European Workshop on Knowledge Acquisition for Knowledge-Based Systems. Toulouse e Caylus, França.

Anais… Berlim, Alemanha: Lecture Notes In Computer Science, Springer-Verlag, 1993, p.340-359.

TOFFOLI, T. Reversible computing. In: DE BAKKER, J. W.; VAN LEEUWEN, J (eds). Automata, Languages and Programming, 7th Colloquium, **Anais…** Lectures Notes in Computer Science, Springer, v.84, p.632-644, 1980.

USCHOLD, M.; GRÜNINGER, M. Ontologies: Principles, Methods and Applications. **Knowledge Engineering Review**, v.11, n.2, p.93-155, 1996. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.111.5903&rep=rep1&type=pdf>. Acesso em 30/06/2010.

_____.; KING, M. Towards a Methodology for Building Ontologies. In: SKUCE, D. (Ed.). **IJCAI'95 Workshop on Basic Ontological Issues in Knowledge Sharing**. Montreal, Canadá, p.6.1-6.10, 1995. Disponível em: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.55.5357&rep=rep1&type=pdf. Acesso em 30/06/2010.

_____.; JASPER, R. A Framework for Understanding and Classifying Ontology Applications. In: BENJAMINS, V. R. (Ed.) IJCAI-99 Workshop on Ontologies and Problem-SolvingMethods (KRR5) Stockholm, Sweden, August 2, 1999. **Proceedings…**CEUR Workshop Proceedings v.18, p.11.1-11.12. Amsterdam, The Netherlands, 1999. http://www.cs.man.ac.uk/~horrocks/Teaching/cs646/Papers/uschold99.pdf. Acesso em 30/06/2010.

USTINOV, A. Quantum Computing using Superconducting Circuits. In: WASER, R. **Nanoelectronics and Information Technology - Advanced Electronic Materials and Novel Devices**. Berlin: Wiley-VCH, 2003.

VAN HEIJST, G., SCHREIBER, A. & WIELINGA, B. Using Explicit Ontologies in KBS development. **International Journal of Human-Computer Studies**, v.46, p.183-292, 1997.

VENTURA, D.; MARTINEZ, T. Quantum Associative Memory. **Information Sciences**, v.124, n.1-4, p.273-296, 2000.

_____. An Artificial Neuron with Quantum Mechanical Properties. **Proceedings of the International Conference on Artificial Neural Networks and Genetic Algorithms**, p. 482-485, abr. 1997.

VIZZOTO, J.; da ROCHA COSTA, A. C. Linguagens de Programação Quântica - Um Apanhado Geral. **Mini-curso do 1o. WEICQ - Worshop Escola de Informacão e Computacão Quântica**, Pelotas-RS, 2006.

WANG, H.; NEZICH, D.; KONG, J.; PALACIOS, T. Graphene Frequency Multipliers. **IEEE Electron Device Letters**, v.30, n.5, p.547-549, mai. 2009.

WEIS, C. D.; SCHUH, A.; BATRA, A.; PERSAUD, A.; RANGELOW, I. W.; BOKOR, J.; LO, C. C.; CABRINI, S.; SIDERAS-HADDAD, E.; FUCHS, G. D.; HANSON, R.; AWSCHALOM, D. D.; SCHENKEL, T. **Single atom doping for quantum device development in diamond and silicon**. Journal of Vacuum Science and Technology B, v.26, n.6, p.2596-2600, 2008.

WELTY, C.; LEHMANN, F.; GRUNINGER, G.; USCHOLD, M. **Ontology: Expert Systems All Over Again?** Invited panel at AAAI-99: The National Conference on Artificial Intelligence. Austin, Texas, 1999.

WILKINSON, K.; REYNOLDS, D.; SEABORNE, A. **Jena position paper**. 2008. Disponível em: <http://www.w3.org/2001/sw/Europe/events/20031113-storage/positions/hp-1.html. Acesso em 08/08/2008>. Acesso em: 30/06/2010.

WOOTERS, W. K.; ZUREK, W. H.. A single quantum cannot be cloned. **Nature**, v.299, p.802-803, out. 1982.

ZALKA, C. Grover's quantum searching algorithm is optimal. **Physical Review A**, v.60, n.4, p.2746-2751, 1999.

ZEILINGER, A. **A face oculta da natureza: o novo mundo da física quântica**. São Paulo: Ed.Globo, 2005.

ZHANG, Q.; TAKESUE, H. ; NAM, S.W.; LANGROCK, C.; XIE, X.; BAEK, B.; FEJER, M. M.; YAMAMOTO, Y. Distribution of Time-Energy Entanglement over 100 km fiber using superconducting single-photon detectors**. Optic Express,** v.16, n.8, p.5776-5781, 2008.

ZHAO, R.; DUDIN, Y. O.; JENKINS, S. D.; CAMPBELL, C. J.; MATSUKEVICH, D. N.; KENNEDY, T. A. B. ; KUZMICH, A. Long-lived quantum memory**. Nature Physics**, v.5, n.2, p.100-104, 2009.

ZOU, Y.; FININ, T.; CHEN, H. F-OWL: An Inference Engine for Semantic Web. In: **Formal Approaches to Agent-Based Systems**, Springer Berlin / Heidelberg, p.238-248, 2005.

ZUREK, W. H. Decoherence and the Transition from Quantum to Classical - Revisited. **Los Alamos Science**, n.27 p.86-109, 2002.

_____. Decoherence and the Transition from Quantum to Classical. **Physics Today**, v.44, n.10, p.36-44, 1991.

APÊNDICE A - HARDWARE QUÂNTICO

Deutsch (2003, p.8) afirma que os computadores quânticos são muito mais difíceis de implementar do que computadores clássicos, sendo o problema quase o oposto: não se refere à montagem de sistemas compostos, precisamente definidos, para utilizar como componentes, mas isolar os sistemas físicos mais simples que existem na natureza dos sistemas complexos existentes no ambiente. Conseguindo-se isto, deve existir um caminho para permitir pares arbitrários (de q-bits) interagir entre si, de alguma forma, um com o outro. Uma vez que isto é alcançado num certo tipo sistema físico, nenhuma configuração ou equipamentos adicionais são necessários, porque as interações dos sistemas quânticos subjacentes à matéria serão sempre universalmente computacionais.

A teoria da Computação Quântica foi concebida a partir de elementos básicos inspirados em certos experimentos tais como o da garrafa de Stern-Gerlach (EISBERG e RESNICK, 1979; CARUSO e OGURI, 2006). A composição dos blocos básicos como portas quânticas em circuitos mais complexos permitem, por sua vez, operações quânticas mais complexas. Entretanto, a realização prática de tais circuitos é uma das tarefas mais desafiadoras atualmente para os pesquisadores. Várias são as propostas para tornar realidade os circuitos quânticos, porém basicamente girando em torno do uso de fótons, da captura de íons ou pela manipulação da propriedade de *spins* dos átomos.

DiVicenzo (1996) aponta cinco requisitos vitais para que um sistema quântico seja implementado num computador quântico:

1) **Adequação dos graus de liberdade ao espaço de Hilbert**: os graus de liberdade requeridos para manter os dados e executar a computação devem estar disponíveis como dimensões do espaço de Hilbert de um sistema quântico, sendo que este sistema deve ser mais ou menos isolado do seu ambiente.

2) **Confiabilidade do estado inicial**: deve ser possível colocar o sistema quântico em um estado inicial confiável. Este estado deve ser bastante simples, tal como “todos os spins para baixo”, se os q-bits são representados como spins. Portanto, este requisito deve

ser satisfeito se existe a possibilidade de se resfriar o sistema para o seu estado mais baixo (*ground state*).

3) **Isolamento do ambiente**: o sistema quântico a ser utilizado como um computador quântico deve possuir um alto grau de isolamento com o seu ambiente. Este requisito de isolamento está relacionado com a precisão requerida na Computação Quântica.

4) **Controle sobre seqüências de operações quânticas**: Deve ser possível ao sistema quântico a operação de uma seqüência controlada de transformações unitárias, pois todos os algoritmos quânticos são expressos em termos destas seqüências. Também é requerido que estas transformações unitárias possam ser feitas agindo sobre pares específicos ou pequenos grupos de q-bits.

5) **Estabelecimento de uma forma de medida**: por último, é necessário que o sistema quântico contenha uma forma "forte" de medida. O termo "forte" se refere à estados ortonormais de algum operador hermitiano que o sistema quântico contém, projetando a função de onda do sistema de forma irreversível para um das possíveis auto-estados. (DIVICENZO, 1996; PRESKILL, 1998; PATEL,1999, p.13).

A seguir, são apresentados vários tipos de implementações físicas para a Computação Quântica, que tentam atender os requisitos afirmados anteriormente.

1) **Fótons**: o uso de fótons para representar um sistema físico de q-bits é muito atraente. Fótons não possuem carga e interagem muito pouco uns com os outros, podem ser guiados a longas distâncias com perda muito pequena em fibras ópticas, e podem ser manipulados eficientemente e combinados utilizando-se divisores de feixes. Feixes de laser podem ser atenuados até que apenas um fóton possa ser produzido com alta probabilidade, e podem ser detectados com alta eficiência quântica em uma larga faixa de comprimentos de onda (NIELSEN e CHUANG, 2004). Uma das desvantagens da computação quântica utilizando fótons é a dificuldade de se fazê-los interagirem. Algoritmos quânticos longos exigiriam a construção de múltiplos interferômetros interligados, e o alinhamento e estabilização entre eles seriam muito difíceis de obter. Apesar disto, a computação óptica quântica parece ser

promissora no uso da comunicação para a informação quântica, devido ao alto rendimento em uma linha de transmissão utilizando fibra óptica (NIELSEN e CHUANG, 2005). Experimentos demonstrando o emaranhamento de três fótons separados espacialmente já foram observados, e emaranhamentos de mais de duas partículas, os chamados estados correlacionados GHZ (iniciais dos seus descobridores Greenberg, Horner e Zeilinger), são evidenciados aqui, mostrando o papel que a Mecânica Quântica pode ter para os arranjos de comunicação quântica (BOUWMEESTER et al, 1999). O emaranhamento com fótons tem sido largamente estudado e demonstrado, em testes na verificação da presença desta propriedade em sistemas de chave quântica distribuída a distâncias próximas de 100 km, utilizando fibra ótica e detectores de fótons isolados (HONJO et al, 2008; ZHANG et al, 2008).

2) **Armadilhas de íons**: a montagem de computadores quânticos utilizando armadilhas iônicas faz com que os q-bits sejam transportados através de íons presos em fileiras, com os estados quânticos simulando os estados $|0\rangle$ e $|1\rangle$, sendo seu movimento manipuláveis através de feixes de laser pulsados. Como os íons de mesma carga ficam separados através de força de repulsão coulombiana, eles estão suficientemente separados de forma que podem ser endereçados individualmente pelos lasers (PRESKILL, 1998). Uma fila de íons é, portanto, confinada por uma combinação de oscilação e campos elétricos estáticos em alto vácuo. Um simples feixe de laser é dividido por divisores de feixe, tal como os utilizados na implementação óptica, em muitos pares que incidem sobre cada íon. Cada íon possui dois estados de longa vida (por exemplo, diferentes níveis do estado de repouso da estrutura hiperfina; o tempo de vida de tais estados contra o decaimento espontâneo pode levar milhares de anos). Tais estados são ortogonais, sendo possível a representação de q-bits. O conjunto requer interações entre os íons, e isto pode ser fornecido pela propriedade da repulsão Coulombiana. Cirac e Zoller (1995) foram alguns dos precursores do uso de armadilhas de íons e forneceram ideias de como as interações pudessem ser tratadas. Uma série de algoritmos quânticos foi testada utilizando-se armadilhas iônicas, inclusive experimentos de teleporte quântico mostrando a comunicação por emaranhamento com íons de Berílio confinados, alcançando alta fidelidade sobre métodos clássicos (BARRET et al, 2004). A codificação superdensa também foi implementada com dois átomos de Berílio confinados, e o protocolo das quatro operações através dos estados de Bell foi obtido, aumentando-se

a capacidade do canal (SCHAETZ et al, 2004). A porta CNOT também foi simulada de forma eficiente, utilizando-se uma armadilha confinando um único átomo de Berílio, com a dinâmica condicional da porta dependendo somente da comparação entre o tamanho do pacote de onda com o comprimento de onda do feixe incidente (DEMARCO et al, 2002).

3) **RMN – Ressonância Magnética Nuclear**: aqui, os q-bits estão representados por spins em uma determinada molécula em particular. Cada spin pode estar alinhado ou anti-alinhado com um campo magnético constante aplicado. Os spins levam um tempo muito grande para entrar em decoerência, podendo manter seus estados quânticos por um longo período (PRESKILL, 1998). Podem ser aplicados campos magnéticos rotativos pulsados com uma freqüência específica, e induzir oscilações nos spins. Temporizando o pulso de forma suave, pode-se efetuar uma transformação unitária a um determinado spin. Todos os spins da molécula são expostos ao campo rotativo, porém respondem apenas aqueles que entram em ressonância. Os sistemas NMR operam em temperaturas mais quentes que os outros modelos, sendo mais suscetíveis ao ruído, devido às fortes flutuações térmicas aleatórias. Porém, a vantagem é que a computação em NMR é feita em paralelo, sobre uma amostra macroscópica contendo aproximadamente 10^{23} moléculas ou "computadores". A medida é feita de forma que o dispositivo gere uma média (PRESKILL, 1998). Computadores quânticos baseados em NMR têm sido utilizados para demonstrar controle de até 7 q-bits (LAFLAMME et al, 2002) e chegando inclusive a computadores quânticos com 12 q-bits para uso de métodos de controle quântico de benchmarking (NEGREVERGNE et al, 2006).

4) **Cavidades de Eletrodinâmica Quântica (QED)**: uma abordagem para a Computação Quântica que combina o acoplamento de um único átomo com poucos modos ópticos é o uso da eletrodinâmica quântica de cavidades. Uma cavidade pode possuir um alto "fator de qualidade" de forma a existirem um ou dois modos de vibração eletromagnética na cavidade tendo altos valores de intensidade de campo elétrico. Este método é uma solução potencial para o problema apresentado pelo uso de fótons na Computação Quântica Óptica, os quais são bons meios de transporte da informação, porém não conseguem mudar seus estados quânticos a partir de interações um com o outro. Átomos isolados nestas cavidades especiais conseguem suprir esta deficiência com baixo índice de descoerência, e permitem a

combinação de dispositivos para Computação Quântica que unem átomos e fótons. A informação utilizando estas técnicas pode ser representada por estados de fótons, que utilizam átomos para as interações entre eles; ou representadas nos átomos, usando-se fótons para a comunicação entre eles (NIELSEN e CHUANG, 2004).

5) **SQUID – Dispositivo Supercondutor de Interferência Quântica**: é uma das tecnologias candidatas a realização física da Computação Quântica. SQUIDs são dispositivos com alta sensibilidade empregados para medida não destrutiva de campos magnéticos, sendo aplicado nas áreas de biofísica e tecnologia de materiais. O dispositivo é composto de um anel de metal que, resfriado a baixíssimas temperaturas, torna-se supercondutor, separado por finas barreiras de material não supercondutor, formando uma junção *Josephson*. Este anel, também chamado de *loop* supercondutor, permite a circulação de corrente sem resistência, gerando um fluxo magnético quantizado. O efeito túnel de elétrons através desta barreira pode permitir a medida de flutuações de campo eletromagnético na ordem de 10^{-15} tesla (algo em torno de 10 a 11 vezes menor que o campo magnético da Terra) (ALTMAN, 2003). A construção de computadores quânticos utilizando-se supercondutores apresenta uma grande flexibilidade em função do uso de tecnologias padrão para construção de circuitos integrados (USTINOV, 2003).

6) **Pontos quânticos (Quantum dots)**: podem ser considerados como caixas tridimensionais, fabricadas com materiais semicondutores, com potenciais eletrostáticos capazes de confinar um quantum de carga, como um elétron. Sendo uma abordagem de estado sólido, consistem numa nano-estrutura que não excederia 1 μm em cada direção espacial, com tamanhos típicos variando entre 10 nm a 100 nm. O potencial criado no ponto quântico limita o movimento da carga nas três dimensões, manifestando-se apenas em níveis discretos de energia (ADAMOWSKI et al, 2005). A representação de q-bits poderia ser feita mediante a localização de uma carga num ponto ou outro, ou tendo-se os dois estados $|0\rangle$ e $|1\rangle$ em um único ponto quântico. Assim como no caso dos fótons, as operações sobre os estados dos q-bits podem ser implementadas por portas eletrostáticas, tal como os deslocadores de fase, e acopladores de guias de onda, com efeito similar aos divisores de feixe vistos na abordagem de fótons (NIELSEN e CHUANG, 2004). Adamowski et al (2005) ressaltam que tais propriedades das cargas nos pontos quânticos podem ser modificadas e controladas facilmente com os dispositivos eletrônicos modernos. Pontos quânticos também podem

ser utilizados mediante operações de leitura e escrita de q-bits com os *spins* dos elétrons confinados (ADAMOWSKI et al, 2005). Implementações de portas quânticas de um e dois q-bits utilizando estados de spin de elétrons acoplados foram propostas (LOSS e DIVICENZO, 1998), assim como a porta de acoplamento de spin *J* utilizando dois pontos quânticos semicondutores acoplados, sendo que o spin dos elétrons pode ser aproveitado para fazer a Computação Quântica (BURKARD et al, 1999). O uso de pontos quânticos na Computação Quântica se beneficiaria também da tecnologia para a fabricação de semicondutores na escala existente atualmente. Pontos quânticos também são denominados de *átomos artificiais*, e dois ou mais pontos quânticos acoplados por uma barreira de tunelamento formam uma *molécula artificial* (ADAMOWSKI et al, 2005).

7) **Técnicas de Spintrônica**: técnica já é utilizada para a confecção de cabeças de leitura de discos rígidos e memória não-volátil, que pode muito bem ser utilizada na construção de computadores quânticos. Através desta técnica de estado sólido, o controle coerente de *spins* eletrônicos e nucleares em semicondutores é experimentalmente factível (LEUENBERGER et al, 2002). Awschalom et al (2002, 2007) sugerem o uso de diamantes dopados com impurezas, tais como na construção dos semicondutores utilizados em transistores nos chips de processadores. Esta impureza, denominada de N-V, consistiria em átomos de nitrogênio presentes na estrutura cristalina de carbono na forma de diamante. A excitação de um laser sobre um átomo de carbono vizinho pode emitir um fóton que inverte o spin do átomo de nitrogênio. Com um segundo átomo de nitrogênio próximo ao conjunto, tem-se a presença de dois q-bits e a configuração de uma porta lógica quântica C-NOT (não controlado). Radiação eletromagnética ajustada a uma freqüência bem específica pode inverter os spins de acordo com as regras para a porta C-NOT. A vantagem desta técnica é a computação quântica poder ser realizada à temperatura ambiente e o pequeno nível de ruído presente. Estudos atuais consideram o aprimoramento de técnicas para dopagem de átomos com alta precisão como oportunidade para construção de dispositivos que manipulem os estados quânticos de átomos simples (WEIS et al, 2008).

8) **Outras Implementações**: Além das implementações físicas explanadas, outras propostas tem sido estudadas em termos de hardware quântico. Tejada et al (2000) propuseram o uso de partículas magnéticas (aglomerados anisotrópicos, tais como Fe_8) de escala nanométrica de alto valor de spin, utilizando o estado de repouso e o primeiro estado de

excitação de spin, ou uma alternativa baseada em tunelamento; o emaranhamento seria obtido através de conexões dos aglomerados anisotrópicos com linhas de supercondutores com junções Josephson entre eles. Projetos alternativos para a Computação Quântica têm surgido, tal como a proposta do computador quântico topológico. Este tipo de computação se baseia em propriedades topológicas que não se alteram frente a perturbações, vindo daí sua maior vantagem. Esta computação topológica faria uso de partículas ainda não comprovadas chamadas *anyons*, que são bem denominadas de *quasipartículas*. Estas partículas são diferentes dos prótons e elétrons, assemelhando-se mais àquelas encontradas na física de altas energias. A ideia á baseada na suposição de que as partículas são linhas ou cordas que tem um comprimento conforme o movimento da partícula ao longo do tempo, e sua espessura dada pelas dimensões físicas. Pares de *anyons* seriam "trançados", e a computação quântica seria codificada neste entrançamento, de forma que perturbações de ordem eletromagnética não afetariam o estado quântico. Superfícies bidimensionais contendo pares de *anyons* tornariam factíveis os computadores quânticos (COLLINS, 2006). O Quadro 23 mostra algumas pesquisas mais recentes, com potencial para implementação de computadores quânticos robustos.

Pesquisa	Descrição	Autor(es)
Gerador de fótons emaranhados de maior rendimento	Canhão de fótons que gera pares de fótons entrelaçados a partir de pontos quânticos, o que pode permitir a fabricação de diodos eletroluminescentes que emitam pares de fótons a 1 GHz, com possibilidade de incrementar a comunicação quântica e constituir um componente chave para processadores quânticos.	Dousse et al (2010)
Gargalos de spin com pontos quânticos duplos e tunelamento ressonante	Uma molécula artificial construída com componentes semicondutores em nanoescala consegue aprisionar elétrons em bandas discretas de energia,	Huang et al (2010)

	tal como em átomos naturais. Isto permite o controle de spin de elétrons individuais.	
Arquitetura híbrida de átomos e moléculas	Quebra das moléculas de um fio supercondutor em seus átomos constituintes a partir de feixes de laser, sem comprometimento dos dados codificados nestas moléculas, permitindo o controle de q-bits em grupo.	Kuznetsova et al (2010)
Nanotransistor com 7 átomos de comprimento	Construído com átomos de fósforo sobre um cristal de silício, mostrando a possibilidade de construção de dispositivos para a computação quântica com a tecnologia convencional da indústria de semicondutores.	Fuechsle et al (2010)
Processador quântico de dois q-bits programável	Demonstração de um processador quântico que pode ser programado com 15 entradas clássicas para realizar transformações arbitrárias sobre dois q-bits que são gravados em íons aprisionados. Pelo uso de estados quânticos e tomografia de processos, até 160 operações aleatoriamente escolhidas podem ser implementado. O conceito de processadores programáveis estão em concordância com o conceito de um registrador multi-qubit que poderia ser colocado em um núcleo de um processador quântico de larga escala.	Hanneke et al (2010)
Estados	Geração de estados hiperemaranhados com alta	Gao et al (2010)

hiperemaranhados	precisão, contendo seis, oito e dez q-bits, através de polarização e graus de liberdade de momento de fótons, podendo fornecer um versátil meio de teste para aplicações quânticas.	
Controle por corrente elétrica de orientação de spins	Uso de microscópio de tunelamento spin-polarizado para bombeamento de spins de elétrons de átomos em superfícies em estados altamente excitados e captar a orientação espacial resultante do spin. Este controle culmina em uma completa inversão da população de estados de spin e proporciona acesso experimental aos tempos de relaxação de spins de cada estado excitado.	Loth et al (2010)
Confiabilidade de portas quânticas através de tomografia de estados quânticos	Uso de tomografia de processos quânticos para caracterizar totalmente a performance de uma porta quântica universal emaranhada com dois q-bits supercondutores, sendo a primeira demonstração de estado sólido.	Bialczak et al (2010)
Memórias quânticas de longa vida	Tempo de vida excedendo 6 ms de memórias quânticas, mediante o uso de transições de estados de átomos de rubídio confinados em um reticulado ótico linear. Constitui-se num passo em direção à realização de redes quânticas de longa distância e a produção controlada de estados	Zhao et al (2009)

	emaranhados envolvendo átomos e luz.	
Rotações óticas ultrarápidas de spin eletrônico em ponto quântico	Uso de um coletivo de spins de elétrons em pontos quânticos focalizados em um pequeno número de modos de precessão sob um campo magnético por bombeamento ótico periódico, com pulsos na escala de picosegundos. Spins de elétrons em pontos quânticos são particularmente atrativos para implementação de q-bits.	Greilich et al (2009)
Teletransporte de funções de onda sobre distâncias macroscópicas	Demonstração do controle coerente da extensão espacial de uma função de onda atômica por expansão e contração da função de onda sobre uma distância de mais de um milímetro. O processo de coerência quântica é totalmente determinístico, reversível e em concordância quantitativa com um modelo analítico.	Alberti et al (2009)
Correção quântica de erros	Implementação experimental de correção quântica de erros para informação quântica codificada em variáveis contínuas, baseadas em emaranhamento de nove feixes óticos, constituindo-se nuam adaptação do esquema original de nove q-bits de Shor permitindo total correção quântica de erros contra o erro de um feixe arbitrário.	Aoki et al (2009)

Teleporte quântico entre átomos	Teleporte de informação quântica entre dois átomos separados por um metro de distância, com potencial para uso em 'repetidores quânticos' em grande escala e capaz de funcionar como uma rede de memórias quânticas.	Olmschenk et al (2009)
Computador quântico de 2 q-bits de estado sólido	Desenvolvimento de um processador quântico de estado sólido, com os q-bits trocando dados através da luz e os estados mantendo a coerência em cerca de 1 microsegundo.	DiCarlo et al (2009)
Operações sucessivas em processador quântico	Experimento demonstrando cálculos computacionais utilizando q-bits iônicos de forma sustentável, mantendo os dados armazenados mesmo depois de lidos.	Home et al (2009)
Fios de diamante para interligação de circuitos quânticos	Canais de dimensões da ordem de micrômetros tornam possível a condução de fótons. Divisores de feixe podem ser montados e possibilitar a computação quântica com luz.	Hiscocks et al (2008)
Ampliação do tempo de duração do spin do elétron	Uso de microondas para controle de spin de elétrons em um chip de silício polarizado com dois eletrodos, sob um forte campo magnético, com o valor do spin sendo medido pela corrente elétrica fluindo entre os eletrodos.	McCamey et al (2008)

Quadro 23: Trabalhos recentes na área de hardware para computação quântica.
Fonte: Elaborado pelo autor.

APÊNDICE B – ALGORITMO SIMULADO DE *MERGING*

O algoritmo para união de ontologias precisa encontrar as classes que são comuns às ontologias em questão. Um circuito quântico com 9 q-bits foi construído e seu processamento executado no simulador de forma a buscar classes comuns, conforme mostrado na Figura 57. Este circuito envolve a recuperação da superposição das classes em memórias distintas, para que o oráculo faça a comparação e modifique o sinal da amplitude do estado desejado (MEDEIROS et al, 2010). Os registradores são simples, contendo 2^2=4 endereços de memória, com os dados representados com resolução 2^2=4, assumindo assim valores na faixa [0,3]. A primeira memória quântica tem os valores-exemplo [0,1,**2**,1] e a segunda memória quântica os valores-exemplo [3,**2**,3,3]. Assim, a classe comum entre as ontologias é a classe representada pelo estado **2**. O circuito inicia no estado $|0\rangle$. As fases do algoritmo são apresentadas a seguir, com os valores na representação de bit sendo visualizados no Quadro 24:

1) **Aplicação da Porta Walsh-Hadamard** aos q-bits 8 e 9, produzindo uma superposição de 4 estados. Estes quatro estados são necessários para o endereçamento da primeira memória quântica;

2) **Aplicação da Porta Walsh-Hadamard** aos q-bits 4 e 5, sendo gerada agora uma superposição de 16 estados; Os estados em superposição para os q-bits 4 e 5 agora endereçam a segunda memória quântica.

3) **Recuperação dos dados da primeira memória quântica**, os quais são colocados nos q-bits 6 e 7. No quadro, o endereço está representado em verde e os dados em amarelo;

4) **Recuperação dos dados da segunda memória quântica**, os quais são colocados nos q-bits 2 e 3. Novamente, no quadro, o endereço está representado em verde e os dados em amarelo;

5) **Comparação dos q-bits 2 e 6 com uma porta C-NOT**. Se o q-bit 6 tiver o valor 1, o q-bit 2 terá seu valor invertido; O objetivo é que, se os q-bits forem iguais, retornarão zero no q-bit 2.

6) **Comparação dos q-bits 3 e 7 com uma porta C-NOT**. Se o q-bit 7 tiver o valor 1, o q-bit 3 terá seu valor invertido; O objetivo é que, se os q-bits forem iguais, retornarão zero no q-bit 3.

7) **Aplicação da porta Toffoli**, com as entradas invertidas (portas de controle vazadas, ou seja, os q-bits 2 e 3). O significado é que, se ambos forem zero (os estados comparados são iguais), o q-bit 1 terá seu valor invertido, ou seja, assumirá o valor 1. No Quadro 24, a posição do q-bit 1 está em azul;

8) **Aplicação da porta Z** ao q-bit 1, invertendo o sinal da amplitude para o estado cujo q-bit assumiu valor 1. O sinal das amplitudes está marcado em cor azul;

9) **Troca dos valores dos q-bits 4 e 6**. O objetivo é agrupar os q-bits de endereçamento

10) **Troca dos valores dos q-bits 5 e 7**. Com os q-bits de endereçamento agrupados, marcados em amarelo, pode-se ver que abrangem a faixa [0,15].

Com os q-bits de 6 a 9, têm-se os endereços necessários, e juntamente com o sinal modificado da amplitude buscada, deve ser aplicada a contagem quântica (para determinação da quantidade de classes comuns) ou ainda a busca de Grover (para encontrar estocasticamente as classes comuns) sobre estes q-bits, que irá amplificar a amplitude do estado "marcado". Os outros q-bits são descartados. Pode-se manter os estados originais dos q-bits se forem colocadas portas adicionais no circuito de maneira simétrica, de acordo com cada porta que foi colocada após às memórias quânticas, em posição invertida no sentido da realização do circuito (exceto a porta Z). Assim, este circuito foi simulado produzindo os resultados que mostram no Quadro 24, e pode-se ser ampliado para endereçar mais estados de memória e valores de dados de maior resolução, aumentando o número de q-bits correspondentes, o número de portas C-NOT para comparação e o número de portas de troca para agrupamento dos q-bits de endereço. O total de q-bits para o circuito é dado então pela seguinte expressão:

$$N=A_{QRAM1} + A_{QRAM2} + D_{QRAM1} + D_{QRAM2} + 1$$

onde A_{QRAM1} e A_{QRAM2} são o número de q-bits para o endereçamento dos estados, e D_{QRAM1} e D_{QRAM2} são o número de q-bits para a representação dos dados endereçados. Um q-bit adicional deve ser colocado para a porta Z. Como exemplo, para endereçar até 64 classes, com até 64 valores diferentes, seriam necessários N=25 q-bits para o circuito.

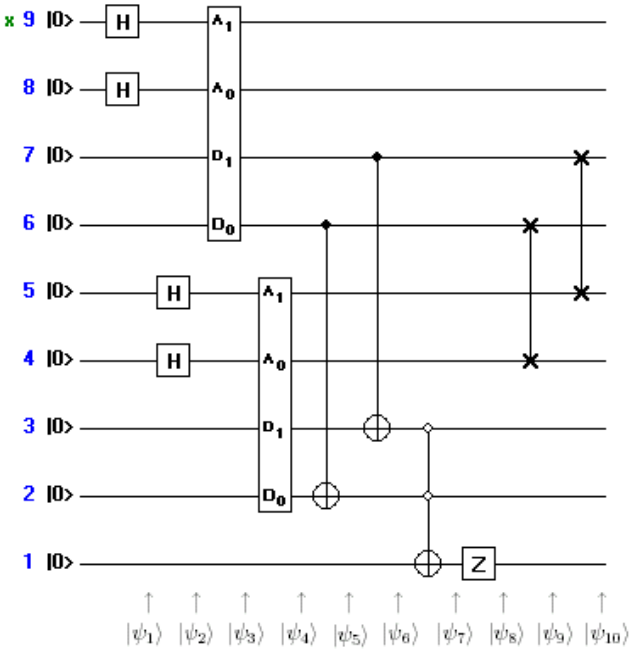


Figura 57: Circuito quântico para o *merging* de ontologias.
Fonte: Elaborado pelo autor.

$\|\psi_1\rangle$ 987654321	$\|\psi_2\rangle$ 987654321	$\|\psi_3\rangle$ 987654321	$\|\psi_4\rangle$ 987654321	$\|\psi_5\rangle$ 987654321
+|000000000>	+|000000000>	+|000000000>	+|000000110>	+|000000110>
+|010000000>	+|000001000>	+|000001000>	+|000001100>	+|000001100>
+|100000000>	+|000010000>	+|000010000>	+|000010110>	+|000010110>
+|110000000>	+|000011000>	+|000011000>	+|000011110>	+|000011110>
	+|010000000>	+|010100000>	+|010100110>	+|010100100>
	+|010001000>	+|010101000>	+|010101100>	+|010101110>
	+|010010000>	+|010110000>	+|010110110>	+|010110100>
	+|010011000>	+|010111000>	+|010111110>	+|010111100>
	+|100000000>	+|101000000>	+|101000110>	+|101000110>
	+|100001000>	+|101001000>	+|101001100>	+|101001100>
	+|100010000>	+|101010000>	+|101010110>	+|101010110>
	+|100011000>	+|101011000>	+|101011110>	+|101011110>
	+|110000000>	+|110100000>	+|110100110>	+|110100100>
	+|110001000>	+|110101000>	+|110101100>	+|110101110>
	+|110010000>	+|110110000>	+|110110110>	+|110110100>
	+|110011000>	+|110111000>	+|110111110>	+|110111100>
$\|\psi_6\rangle$	$\|\psi_7\rangle$	$\|\psi_8\rangle$	$\|\psi_9\rangle$	$\|\psi_{10}\rangle$

987654321	987654321	987654321	987654321	987654321
+|000000110>	+|000000110>	+|000000110>	+|000000110>	+|000000110>
+|000001100>	+|000001100>	+|000001100>	+|000010110>	+|000100100>
+|000010110>	+|000010110>	+|000010110>	+|000100100>	+|001000110>
+|000011110>	+|000011110>	+|000011110>	+|000110110>	+|001100110>
+|010100100>	+|010100100>	+|010100100>	+|010001100>	+|010001100>
+|010101110>	+|010101110>	+|010101110>	+|010011100>	+|010101110>
+|010110100>	+|010110100>	+|010110100>	+|010101110>	+|011001100>
+|010111100>	+|010111100>	+|010111100>	+|010111100>	+|011101100>
+|101000010>	+|101000010>	+|101000010>	+|101000010>	+|100010010>
+|101001000>	+|101001001>	-|101001001>	+|101010010>	-|100110001>
+|101010010>	+|101010010>	+|101010010>	-|101100001>	+|101010010>
+|101011010>	+|101011010>	+|101011010>	+|101110010>	+|101110010>
+|110100100>	+|110100100>	+|110100100>	+|110001100>	+|110001100>
+|110101110>	+|110101110>	+|110101110>	+|110011100>	+|110101110>
+|110110100>	+|110110100>	+|110110100>	+|110101110>	+|111001100>
+|110111100>	+|110111100>	+|110111100>	+|110111100>	+|111101100>

Quadro 24: Evolução dos estados do circuito de *merging* simulado.
Fonte: Elaborado pelo autor.

APÊNDICE C – ALGORITMOS CONVENCIONAIS

Este apêndice descreve alguns algoritmos que são utilizados de forma convencional para tratar os problemas abordados pelos algoritmos quânticos. Com isso, busca-se auxiliar o estudo da complexidade dos algoritmos em cada caso. Todos os algoritmos explicados aqui estão escritos em linguagem C.

C.1 ALGORITMO DE VALIDAÇÃO

O algoritmo de validação de instâncias precisa testar a cláusula que contém duas comparações: a igualdade entre clientes e a desigualdade entre vendedores. Considerando que as instâncias já estejam carregadas na memória, em um vetor m, o algoritmo no Quadro 25 faz a tarefa de validação. A variável contadora k conterá o número de instâncias inválidas. Os dois loops "for" encadeados mostram a complexidade $O(N^2)$ do algoritmo.

```
      /* Algoritmo de validação de instâncias */
      /* N = total de instâncias */
      /* i,j = variáveis dos loops */
      /* k = contador de instâncias inválidas */
      /* m[0..N-1] = vetor com as instâncias já carregadas
da memória

      int i,j,k;
      int m[];
      ...
      k=0;
      for(i=0;i<N;i++)
         for(j=0;j<N;j++)
            if((m[i].c == m[j].c) && (m[i].v != m[j].v))
               k++;
...
```

Quadro 25: Código em linguagem C do algoritmo de validação.
Fonte: Elaborado pelo autor.

Um refinamento deste algoritmo pode ser feito, devido à varredura das instâncias se para as primeiras instâncias, fazendo com que a variável j reinicie da instância indexada por i ao invés de 0. Porém, a complexidade se reduzirá a $O(N^2/2)$, mantendo ainda a característica quadrática. Este algoritmo serve de benchmark para o algoritmo descrito na seção 3.1.

C.2 FECHAMENTO TRANSITIVO

O algoritmo de fechamento transitivo (TENENBAUM et al, 1995) é aplicado para indicar a existência de um caminho ou arco entre dois nós que passa através de um nó de um grafo. Para isso, faz uso de uma matriz de adjacências para calcular os caminhos de comprimento crescente, até que atinja o número máximo de nós existentes em um grafo (TENENBAUM et al, 1995). Na Figura 58 está um exemplo de demonstração desta situação. O Quadro 26 explica o caminho direto dos arcos entre os nós, ou seja, o caminho de comprimento um, através da matriz de adjacências. Nesta matriz, o valor um indica quais nós estão conectados No Quadro 27 está a matriz de adjacências de caminho dois, mostrando os relacionamentos transitivos entre os nós existentes neste grafo.

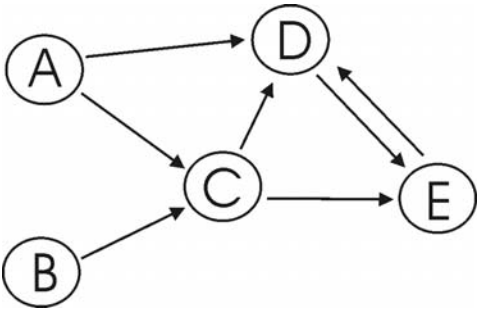


Figura 58: Exemplo de grafo para fechamento transitivo.
Fonte: Adaptado de Tenenbaum et al (1995).

	A	B	C	D	E
A	0	0	1	1	0
B	0	0	1	0	0
C	0	0	0	1	1
D	0	0	0	0	1
E	0	0	0	1	0

Quadro 26: Matriz de adjacência inicial.
Fonte: Adaptado de TENENBAUM et al (1995).

	A	B	C	D	E
A	0	0	0	1	1
B	0	0	0	1	1
C	0	0	0	1	1
D	0	0	0	1	0
E	0	0	0	0	1

Quadro 27: Matriz de adjacência de caminho dois.
Fonte: Adaptado de TENENBAUM et al (1995).

O algoritmo "fechamento_transitivo" (TENENBAUM et al, 1995) descrito no Quadro 28 mostra como as matrizes de adjacências de profundidade crescente são construídas. A matriz de adjacências é a variável "*a*", e a variável "*p*" conterá a combinação "ou" de todas as matrizes de adjacências. A constante "*N*" é o número máximo de nós, e a matriz de adjacências possui, portanto, tamanho *N*x*N*.

```
fechamento_transitivo(a, p)
int a[][N], p[][N];
{
   Int i, j, k;
   Int np[N][N], ap[N][N];

   for(i=0; i<N; ++i)
      for(j=0; j<N; ++j)
         ap[i][j] = p[i][j] = a[i][j];

      for(i=0; i<N; ++i)
      {
         prod(ap,a,np); /* chama sub-rotina prod*/

         for(j=0; j<N; ++j)
            for(k=0; k<N; ++k)
               p[j][k] = p[j][k] || np[j][k];

         for(j=0; j<N; ++j)
            for(k=0; k<N; ++k)
               ap[j][k] = np[j][k];

      }
   }

   prod(a,b,c)
   int a[][N], b[][N], c[][N];
   {
      int i, j, k, val;
```

```
            for(i=0; i<N; ++i)
               for(j=0; j<N; ++j)
               {
                  val = FALSE;
                  for(k=0; k<N; ++k)
                     val = val || (a[i][k] && b[k][j]);
                  c[i][j] = val;
               }
   }
```

Quadro 28: Código em linguagem C para o fechamento transitivo. Fonte: adaptado de TENENBAUM et al (1995).

A eficiência deste algoritmo mostra que a sub-rotina "prod", que é $O(N^3)$, em função dos três loops encadeados, é chamada pela rotina principal dentro de outro loop, configurando assim que o algoritmo completo é $O(N^4)$.

C.3 ALGORITMO DE WARSHALL

Este algoritmo melhora a ineficiência do fechamento transitivo, que reduz em um loop "for" a construção da trilha final pelas matrizes de adjacências ao fazer apenas o loop mais interno caso existe um nó do qual possam partir mais arcos. O algoritmo é descrito no Quadro 29, sendo evidente sua simplicidade em relação ao algoritmo anterior (TENENBAUM et al, 1995).

```
   warshall(a, p)
   int a[][N], p[][N];
   {
      int i, j, k;

      for(i=0; i<N; ++i)
         for(j=0; j<N; ++j)
            p[i][j] = a[i][j];

      for(k=0; k<N; ++k)
         for(i=0; i<N; ++i)
            if(p[i][k] == TRUE)
               for(j=0; j<N; ++j)
                  p[i][j] = p[i][j] || p[k][j];

   }
```

Quadro 29: Código em linguagem C do algoritmo de Warshall. Fonte: adaptado de TENENBAUM et al (1995).

Em função disso, pode-se concluir que a eficiência do algoritmo de Warshall possui eficiência $O(N^3)$. Para o caso de se requerer apenas um caminho de comprimento 2, tal como o utilizado para demonstrar o algoritmo de raciocínio transitivo descrito na seção 3.2, a eficiência irá reduzir-se a $O(N^2)$.

C.4 ALGORITMO PARA MERGING

Quanto ao algoritmo convencional para o benchmark daquele apresentado na seção 3.3, as classes de cada ontologia podem ser colocadas em vetores específicos, comparando-se os nomes entre cada uma para identificar aquelas que são comuns. Diferente do algoritmo quântico, que retorna as classes comuns de forma estocástica, o algortitmo descrito no Quadro 30 é determinístico. Em função dos dois loops "*for*" encadeados, sua eficiência é de $O(N^2)$.

```
      /* Algoritmo de merging */
      /* N = total de instâncias */
      /* i,j = variáveis dos loops */
      /* k = contador de instâncias inválidas */
      /*  onto1[0..N-1]  e  onto2[0..N-1]  =  vetores  com  as
classes */
      /* já carregadas da memória */

      int i,j,k;
      int onto1[], onto2[];
      ...
      (carga das classes nos vetores)
      ...
      k=0;
      for(i=0;i<N;i++)
         for(j=0;j<N;j++)
            if((onto1[i].name == onto2[j].name)
               /* mostra ou armazena a classe */
...
```

Quadro 30: Código em linguagem C do algoritmo de merging.
Fonte: Elaborado pelo autor.